Smart Agricultural Technology 12 (2025) 101408

Contents lists available at ScienceDirect

Smart Agricultural Technology

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

Enhancing consistency and efficiency of tracking multiple individual broiler chickens in group settings

Guoming Li [a,b,c,d,*], Sai Akshitha Reddy Kota [b], Tongshuai Liu [e], Oluwadamilola Moyin Oso [a], Venkat Umesh Chandra Bodempudi [a,c], Mahtab Saeidifar [a,b,c], Ehsan Asali [a,b], Aravind Mandiga [a,b], Jin Lu [b,c], Geng Yuan [b,c], Ahmad Banakar [f]

[a] Department of Poultry Science, The University of Georgia, Athens, GA 30602, USA
[b] School of Computing, The University of Georgia, Athens, GA 30602, USA
[c] Institute for Artificial Intelligence, The University of Georgia, Athens, GA 30602, USA
[d] Institute for Integrative Precision Agriculture, The University of Georgia, Athens, GA 30602, USA
[e] College of Animal Science & Technology, Henan University of Animal Husbandry and Economy, Zhengzhou, Henan 450046, China
[f] Biosystems Engineering Department, Tarbiat Modares University, Tehran 14117-13116, Iran

ARTICLE INFO

Keywords:
Poultry
Tracking
Object detection
Model pruning
Animal welfare
Artificial intelligence

ABSTRACT

Multi-object tracking computer vision algorithms have been progressively researched in precision broiler chicken farming domain. However, the set of algorithms are suffering from identity switches that further downgraded tracking consistency and efficiency due to similar broiler chicken appearance, frequent facility occlusion, and changing environments. The objective of this study was to increase the consistency and efficiency of tracking multiple individual broiler chickens in group settings. The major contributions of this work included data calibration, custom object detection model training for broiler chicken detection, model pruning of the detection models to improve detection efficiency, custom training for broiler chicken feature extraction in tracking algorithms, tracking performance comparison of several multi-object trackers, and kinematic feature extractions of individual broiler chickens after tracking. Videos were collected for 1776 Cobb 500 male broiler chickens. The birds were raised in 48 pens (1.2 m wide × 3.0 m long) and kept until week 7. The You Only Look Once (YOLO) v11x outperformed nine other object detection models and achieved over 0.96 precision, recall, and mAP50 for detecting broiler chickens from weeks 2 to 7. The YOLOv11x with 0.09 pruning ratio of total weights can increase 13.5 fps processing speed while not compromising broiler chicken detection performance, compared to the YOLOv11x with full custom weights. The proposed feature extractor combined with Vision Transformer, ResNet152, and DenseNet201 models outperformed other eight deep learning image classifier and enhanced re-identification performance with 0.956±0.032 cosine similarity value and 0.020±0.007 Euclidean distance value. The kinematics-aware machine learning classifier (integrated with the extra trees classifier) performed better in determining same/different broiler chickens than the other 14 machine learning classifiers, with the highest accuracy (0.917), precision (0.958), recall (0.920), and F1 score (0.939). The modified ByteTrack performed better in multi-object tracking accuracy (MOTA) and precision (MOTP), processing speed, and longest tracking duration than the other five trackers. By putting all components together, the integrated model system achieved 0.904±0.073 MOTA, 0.953±0.057 MOTP, 30.1±3.3 fps processing speed, and 17.3 min (12.4 min in the previous study) longest tracking duration for tracking multiple individual broiler chickens from weeks 2 to 7 and in feeder, drinker, and open areas. The proposed modeling system presented potential in improving multiple individual broiler chicken tracking consistency and efficiency. More generalizable and efficient models need to be researched for achieving robust real-time multiple group-housed broiler chicken tracking.

* Corresponding author.
E-mail address: gmli@uga.edu (G. Li).

https://doi.org/10.1016/j.atech.2025.101408
Received 16 February 2025; Received in revised form 12 August 2025; Accepted 31 August 2025
Available online 1 September 2025

1. Introduction

The U.S. broiler chicken industry continues to grow and expand to support increasing human populations. Despite supplying affordable proteins for humans, the intensified industry is under pressure to provide broiler chickens with larger living spaces. A modern barn accommodates 25,000 to 100, 000 broiler chickens on the open litter floor [27]. The high rearing stocking density is favorable for economic profits but increases the difficulties in inspecting and tracking broiler chickens daily for producers [43]. In a daily basis, broiler chicken producers have to walk through houses and visually observe the flocks based on their own senses to determine their health and well-being situations, such as feathers, behaviors, and sounds. In addition, routine records (e.g., feed and water consumption, mortality rates, and growth rates) can also provide insights into broiler chicken management [44]. While these traditional methods are still important, they are laborious and time-consuming to demonstrate the whole picture of entire flocks and sustain the expanding industry. Manual observation can be time-consuming and labor-intensive and may not always detect subtle signs of stress or illness in a timely manner [4].

Recent advancements in technologies, particularly in areas such as precision livestock farming and artificial intelligence, offer promising solutions to assist producers in managing broiler chicken health and well-being precisely and efficiently [36]. For example, computer vision systems feature non-contact and non-invasive monitoring and tracking of individual broiler chickens. Merenda et al. [40] developed and validated You Only Look Once (YOLO) model and Strong Simple Online and Realtime Tracking (StrongSORT) to track multiple individual broiler chickens and then used machine learning classifiers to classify feeding, drinking, active, and inactive behaviors. They achieved over 90 % accuracy for classifying feeding and drinking behaviors of individual broiler chickens. Neethirajan [42] combined YOLOv5 and Kalman Filtering for tracking individual broiler chickens in backyards. This tracking system's overall performance across all timestamps was summarized with a ground truth total of 310 chickens and a total tracked count of 323, resulting in an over-counting of 13. Li et al. [28] tracked and characterized spatiotemporal and three-dimensional locomotive behaviors of individual broiler chickens using a pose estimation algorithm, DeepLabCut embedded with ResNet101 classification model. The algorithm extracted key points on back of broiler chickens and achieved an accuracy of 100 % for recognizing key points. Li et al. [31] integrated Faster Region-based Convolutional Neural Network (Faster R-CNN) for detecting broiler chickens and Euclidean Distance Tracker to track them continuously walking around a feeder as an indication of restricted feeding behaviors. Additionally, when identifying broiler chickens around the feeder, the model maintained a high F1 score of above 95 %, with precision slightly lower at 92.5–94.5 %, indicating reliable detection across various behaviors near feeders. Doornweerd et al. [15] combined YOLOv7-tiny and SORT for tracking 35 non-color marked broiler chickens in experimental pens and concluded that tracking errors primarily occurred when broiler chickens were occluded by the drinker, and frequently when broiler chickens were in close proximity (within 10 cm), with velocity and acceleration appearing to have a lesser impact on tracking errors. The number of identity (ID) switches per ground-truth trajectory ranged from 5 to 20, with a mean of 9.92. These switches indicated instances where tracking continuity of individual broiler chickens was compromised. Campbell et al. [8] proposed a CNN bird detector, network flow tracker, and component-based feature saliency Gaussian mixture model to track and monitor activity and trajectory of individual broiler chickens in commercial farms. The proposed system was able to track broiler chickens across video frames with a multi-object tracking accuracy (MOTA) of 74.7 %.

The abovementioned video-based tracking, also known as Multi-Object Tracking (MOT) in computer vision, is the task of detecting objects of interest within video frames and associating those detections across consecutive frames in time domain. In the above MOT cases, two paradigms exist: detection-and-tracking (detection and tracking are performed jointly) and tracking-by-detection (tracking are performed after detection). The latter one was more widely adopted, driven by improvements in object detection methods [53]. However, those tracking paradigms failed in consistently tracking individual broiler chickens, for instance IDs were switched within 12 min in the above listed examples due to multiple reasons. Individual broiler chickens exhibit physically and mentally complex, individually different, time-varying, and dynamic characteristics [5]. Those dynamic variations in individual levels make tracking challenging. Given the fact that a broiler chicken is raised in less than 1 ft^2 floor area, the MOT tasks are heavily influenced by occlusions, overlapping, similar appearances among broiler chickens, motion blur due to fast movement, inconsistent illuminations and backgrounds, and low resolutions [29].

Those challenges motivate the team to continuously improve algorithmic framework for maintaining IDs and improving multiple individual broiler chicken tracking accuracy. Custom training can always improve model performance in custom datasets for YOLO object detection models [40]. Tracking models require precise feature maps [6] that are currently built on Common Objects in Context (COCO) dataset [35] and are not suitable to extract broiler chicken features in densely-populated, low-light, and humid environments, thus custom training on feature extractors of tracking schemes being needed. This is also critical for re-identification in MOT tasks [54], in which new IDs are reassigned to the objects of interest once the initial IDs are missed. Kinematic features like Euclidean Distance changes of individual birds have demonstrated great potential for boosting tracking performance [31] but have not been integrated into recent trackers such as DeepSORT [52], StrongSORT [17], and ByteTrack [53]. The two sets (tracking and detection) of models can generate millions of parameters inside, which is not supportive for real-time monitoring. Model pruning balancing tracking accuracy and speed is needed to improve inference efficiency [24].

In sum, the objective of this research was to track multiple individual broiler chickens and maintain their IDs consistently and efficiently in group settings by improving detection, feature extraction, tracking, and model pruning. The developed tracking framework was then applied to extract kinematic characteristics of individual broiler chickens, such as velocity, acceleration, and trajectories.

2. Materials and methods

2.1. Animal housing and video data collection

The study was conducted at the University of Georgia's Poultry Research Center from May to June 2024. The center was located at the southern part of Athens in Georgia State. Two environmentally controlled rooms were used, with each measuring 17.2 m long × 11.4 m wide and consisting of 24 experimental pens. Each room was divided into two rows, and each row consisted of 12 identical pens (1.2 m wide × 3.0 m long for each). Each pen was equipped with two feeders (at both ends) and two drinking lines (in the middle). A total of 1,776 day-old Cobb 500 broiler chickens were randomly assigned into 48 pens, with 37 broiler chickens raised each pen. Feed and water were supplied *ad libitum*, and the broiler chickens were reared under standard environmental conditions following the Cobb management guideline [12]. Broiler chickens were raised up to 49 days (week 7). The light was set as full brightness (20–30 lx) from days 0–9 and as dim (~5 lx) from days 10–49. The temperature was set at 33.9 °C from days 0–6; 31.1 °C from days 7–13; 28.3 °C from days 14–20; 25.6 °C from days 21–27; 22.8 °C from days 28–34; and 21.1 °C from days 35–49. All procedures were approved by Institutional Animal Care and Use Committees at the University of Georgia (protocol number: A2023 07–016-Y1-A0).

A security camera (NHD-887MSB, Swann Security, Santa Fe Springs, CA) was installed on the ceiling of each pen with distance of 3.05 m from the ground. The specifications of the NHD-887MSB cameras (151 mm ×

70 mm × 70 mm) with a shutter control function included the 93° viewing angle, 150 ft/45 m infrared night vision or 200ft/60 m with ambient lights, -30 to 55 °C operating temperature, and 1.8-pound item weight. Broiler chickens were continuously recorded with two 16-channel video recorders (SRDVR-85680H-US, Swann Security, Sacta Fe Springs, CA). The video recording was set at a resolution of 1,024 × 768 pixels at a sample rate of 15 frames per second (fps), and 24 video episodes per hour per pen were recorded on a daily basis. The videos were stored as .MP4 files on a 20-terabyte external hard disk. Recordings were made on four consecutive days of each of weeks 2–7. Week 1 was skipped due to the adaptation and mortality replacement. Fig 1 shows some photos of experimental pens for the study.

2.2. Overall workflow of algorithm developments

The workflow of this research comprised six major components, as illustrated in Fig. 2, including data calibration, custom object detection model training for broiler chicken detection, model pruning of bird detection models to improve detection efficiency, custom training for broiler chicken feature extraction for tracking algorithms, tracking performance comparison of multiple trackers, and kinematic feature extractions of individual broiler chickens after tracking. The main programming language utilized was Python. Key Python libraries included OpenCV and NumPy for image processing, Matplotlib for graphical representation, and Torchvision for feature extraction and model pruning. Additionally, visualization platforms (e.g., TensorBoard) were used for enhancing model performance. The YOLO library was imported from Ultralytics, a company known for its cutting-edge computer vision solutions and advancements in YOLO models. Model training and validation were executed on Google Colab, which provided 12.7 GB of RAM and 16 GB of T4 GPU memory, supported by a dual-core CPU running at 2.30 GHz. The specifications of local computers used for inference and testing involved the processor of Intel® Core™ i7–8665 U CPU @ 1.90 GHz–2.11 GHz, the operation system of 64-bit Windows 11 Pro, and 16 GB RAM.

2.3. Data preprocessing

The frames recorded at nearly 3-m height can have fish-eye effects, in which far-end broiler chickens were much smaller than near-end ones. Therefore, far-end broiler chickens could always have shorter movement trajectories or smaller movement speed than near-end ones due to disproportionate body dimensions in distorted frames. A checkerboard was randomly placed at multiple angles and places of a target pen under a camera during the experiment (Fig. 3). The checkerboard was 40 cm in width and 50 cm in length with 4 grids in width and 5 grids in length. The checkerboard was placed along the edges and in the middle of each of the 48 pens, with different orientations under the static cameras. Videos including checkerboard placement were recorded and converted into frames for distortion correction using OpenCV in Python.

The distortions consisted of two representative kinds: radial distortion [22] and tangential distortion [37]. Radial distortion causes straight lines to appear curved. Radial distortion becomes larger for the farther points from the center of an image. Tangential distortion occurs because the image-taking lens is not aligned perfectly parallel to the imaging plane. So, some areas in the image may look nearer than expected. The amount of radial distortion and tangential distortion can be presented in Eqs. (1) and (2), respectively.

$$x_{distorted} = x\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right),\ y_{distorted} = y\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \tag{1}$$

$$x_{distorted} = x + \left[2p_1 xy + p_2\left(r^2 + 2x^2\right)\right] y_{distorted} = y + \left[p_1\left(r^2 + 2y^2\right) + 2p_2 xy\right] \tag{2}$$

$$r = \sqrt{\left(x_{distorted} - x_c\right)^2 + \left(y_{distorted} - y_c\right)^2} \tag{3}$$

$$Distortion\ coefficients = \left(k_1, k_2, p_1, p_2, k_3\right) \tag{4}$$

where $(x_{distorted},\ y_{distorted})$ are the coordinates captured by cameras; $(x,\ y)$ are the coordinates after distortion correction; $(x_c,\ y_c)$ are the centroid coordinates of a image frame; r is the Euclidean distance between the distorted image point and the distortion center; k_n is the n^{th} radial distortion coefficient; and p_n is the n^{th} tangential distortion coefficient.

In addition to finding the distortion coefficients, some other information like the intrinsic and extrinsic parameters of the camera were needed. Intrinsic parameters are specific to a camera. They include information like focal length $\left(f_x, f_y\right)$ and optical centers $\left(c_x, c_y\right)$ (expressed as a 3×3 matrix in Eq. (5)). The focal length and optical centers can be used to create a camera matrix, which can be used to remove distortion due to the lens of a specific camera. The camera matrix is unique to a specific camera, so once calculated, it can be reused on other images taken by the same camera. Extrinsic parameters correspond to rotation and translation vectors which translate a coordinate of a three-dimensional points into a coordinate system.

Fig. 1. Photos of experimental setup for development of deep learning trackers. Broiler chickens in the photos were 5 weeks old.

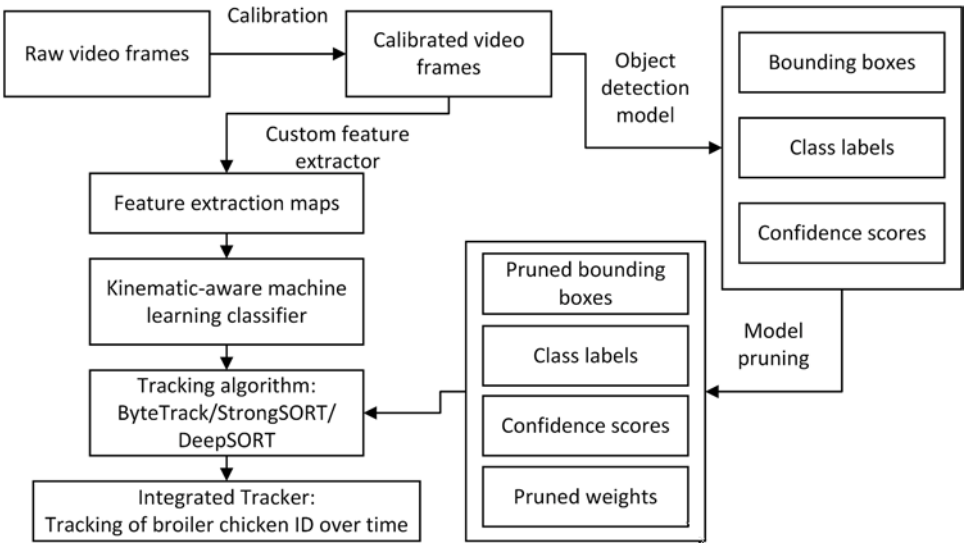

Fig. 2. Workflow diagram of algorithm development for tracking individual broiler chickens.

Fig. 3. Photo of utilizing a checkerboard for calibrating the distortion in image frames.

$$camera\ matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

To find the coefficients and parameters, image thresholding and corner detection algorithms were run on some sample images (e.g., ≥10 images per pen with different locations and orientations) of a well-defined pattern (e.g., checkerboard). Some specific points, in which relative positions (e.g., square corners in the checkerboard) were known, were found. The parameters and coefficients can be obtained by correlating the coordinates of these points in real world space with the coordinates in images. The calibration matrix was obtained by minimizing the reprojection error defined in Eq. (6).

$$RPE_{RMS} = \sqrt{\frac{\sum_i \sum_j \|p_{ij} - \hat{p}_{ij}\|^2}{N}} \quad (6)$$

where *RPE* is reprojection error; *RMS* is root-mean-square; *i* is the point's unique ID on the calibration board; *j* is the pose/image number; *p* is the measured point; and $\hat{p}$ is the projection point obtained by taking the i^{th} point nominal calibration board coordinates, applying the j^{th} pose's extrinsic transformation, and lastly applying the camera's projection mapping.

In addition, three focal birds in each of the 48 pens were selected to evaluate the performance before and after the data preprocessing (camera calibration). Those birds located at the far end of the camera views with distortion effects. Birds were segmented with free segmentation software, Labelme, and the body area of each bird was summed up using OpenCV-Python. The deviation calculation procedure is presented in Eq. (7).

$$Deviation\ (\%) = \frac{|A_{before} - A_{after}|}{A_{before}} \quad (7)$$

where *A* indicates the body area of each bird in pixels; the subscript '*before*' indicates before camera calibration; and the subscript '*after*' indicates after camera calibration. The deviation was calculated for each bird in each pen and averaged on a weekly basis.

After calibration, the parameters included camera matrix that encapsulated focal length and principal points, distortion coefficient matrix modeling radial and tangential lens distortions, and extrinsic parameter matrix consisting of rotation and translation vectors. Those

calibration parameters were stored and used to correct distortion of all video frames for subsequent processing. Due to pen differences, a total of 48 sets of calibration parameters were stored and implemented for corresponding pens for distortion correction. The undistorted images were cropped based on region of interest (ROI) to remove black regions resulting from distortion correction. Further image processing techniques, such as padding to maintain original image dimensions or alpha channel adjustments, were performed for seamless integration into other workflows.

2.4. Custom training for broiler chicken detection

The YOLO models performed well on predicting 80 common objects in the COCO dataset (Table 1). However, based on the preliminary examinations, those models performed poorly (<40 % accuracy) in the broiler chicken image dataset due to lack of data variations in the pretrained dataset. Thus, custom training of YOLO models was needed. YOLOv8 and YOLOv11 models were selected from Ultralytics [50] for customized development for bird detection due to their popularity and large variants of models for object detection, including nano (n), small (s), medium (m), large (l), and extra-large (x) versions. A total of 9188 image frames were converted from the recorded videos of the 48 pens and from weeks 2 to 7 and annotated using the free object detection labeling tool (LabelImg). Each bird was annotated with a bounding box along with the class name, 'chicken', resulting in a total of 339,956 broiler chicken instances. A typical rule-of-thumb in data splitting for dataset sizes from 100 to 10,000 images dictates that images should be split in 70:30 or 80:20 for training and validation set [30]. This general practice of splitting the dataset was followed to ensure sufficient data for training while retaining enough for effective validation. This balanced approach, often seen in studies, helps in model optimization, and prevents overfitting. The annotated dataset was split into 80:20. Models were fine-tuned using optimized hyperparameters: a learning rate of 0.001, a batch size of 16, and training over 100 epochs.

The performance of the bird detection was measured standard object detection metrics, including precision (Eq. (9)), recall (Eq. (10)), F1 score (Eq. (11)), and mAP (Eq. (8)) at IoU threshold of 0.5 (mAP50) and 0.95 (mAP95) [30].

$$mAP = \frac{1}{N}\sum_{i=1}^{N} average\ precision(i) \tag{8}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{9}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{10}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{\text{Precision} + \text{Recall}} \tag{11}$$

$$IoU = \frac{(True\ bounding\ boxes) \cap (Predicted\ bounding\ boxes)}{(True\ bounding\ boxes) \cup (Predicted\ bounding\ boxes)} \tag{12}$$

where i indicates the i^{th} class; IoU is intersection over union; and mAP is mean average precision.

2.5. Model pruning for broiler chicken detection

The YOLO model may rely on over-parametrized modules for identifying multiple broiler chickens. From another aspect, efficient sparse connectivity in biological neural networks can reduce unnecessary parameters and enhance memory or computational space utilization efficiency. Pruning is used to investigate the differences in learning dynamics between over-parameterized and under-parameterized networks [20]. Our approach utilized L1 unstructured pruning, a method known for its efficacy in diminishing model size by selectively zeroing out weights with the smallest absolute values. The weight magnitude-based pruning formulas is presented in Eq. (13). This technique focused on the convolutional layers of the optimal YOLO model after comparison (Section 2.4), which contributed substantially to the model's parameter volume.

$$Prune\ (W,k) = \{w \in\ W\ :|w| \le n^{th}\ smallest(|W|,k) \tag{13}$$

where W is the set of all weights in the model; w is individual weights; and k is the percentage of weights to prune.

Pruning was performed using the PyTorch module, and different percentage (k) of the weights across these layers in a learnable manner, ranging from 0.01 to 0.15 with 0.01 increment. This method maintained the original network architecture, ensuring no modification to the network's structural integrity. The effectiveness of the pruned and integrated model was trained and assessed using the same dataset as mentioned in Section 2.4. Evaluation metrics included Precision (Eq. (9)), Recall (Eq. (10)), and mAP50 and mAP50–95 (Eq. (8)). With increasing proportion of reduced weights, processing speed increased but broiler chicken detection accuracy could decrease. Therefore, the curve of processing speed (fps) versus detection accuracy across the percentages were depicted to determine a sweet point for balancing the two metrics. Once the percentage was completed, the weight reduction was executed by calling the index of specific weights, effectively transforming pruned parameters into regular tensor attributes and thus optimizing the model for efficient operation.

2.6. Strengthening re-identifications of broiler chickens in tracking algorithms

Most of the re-identifications in tracking algorithms rely on the feature maps generated by a pre-trained CNN classifier that are developed from common datasets like ImageNet [14], which is not suitable for broiler chicken farming scenarios as the background, distribution density, and appearance similarity levels are different. The task of this

Table 1
Performance metrics of object detection models pretrained on the COCO datasets.

Model	Size (MB)	Number of layers	Number of parameters	mAP50–95 (%)	Speed-CPU (ms)	GFLOPs
YOLOv8n	6.3	225	3.2M	37.3	80.4	8.7
YOLOv8s	21.5	225	11.2M	44.9	128.4	28.6
YOLOv8m	49.7	295	25.9M	50.2	234.7	78.9
YOLOv8l	83.7	365	43.7M	52.9	375.2	165.2
YOLOv8x	130.0	365	68.2M	53.9	479.1	257.8
YOLOv11n	5.4	319	2.6M	39.5	56.1	6.5
YOLOv11s	18.4	319	9.5M	47.0	90.0	21.5
YOLOv11m	38.7	409	20.1M	51.5	183.2	68.0
YOLOv11l	49.0	631	25.4M	53.4	238.6	86.9
YOLOv11x	109.0	631	57.0M	54.7	462.8	194.9

Note: YOLO = You Only Look Once; COCO = Common Objects in Context; mAP = mean average precision; and GFLOPs = Giga Floating Point Operations Per Second.

section is to strengthen the re-identification of broiler chickens by replacing the pre-trained CNN classifier with custom-built deep learning image classifiers.

Some classical deep learning image classifiers were re-trained as the base models, including DenseNet201 [26], EfficientNetB7 [49], InceptionResNetV2 [48], MobileNetV3Large [25], ResNet152 [23], Vision Transformer (ViT) [16], VGG19 [47], and Xception [11]. These models were loaded with pre-trained weights from ImageNet for raw feature extractions. The last layer of each pre-trained model was unfrozen and fine-tuned with a custom bird dataset, allowing the model to learn domain-specific features from the dataset. In addition, we proposed a merging framework (Fig 4) from the top three classification models among the above-mentioned networks. The merging method was weighted averaging with each of the three feature maps from respective networks (each sharing 1/3 of the weights). The proposed network merged the feature maps from the three models and then used max pooling, activation (ReLU), dropout, flatten, fully connected, and softmax layers to enhance bird classification. Based on results in Section 3.4, the three networks (Vision Transformer, DenseNet201, and ResNet152) had similar cosine similarity (0.918–0.932) and Euclidean distance (0.034–0.113) and outperformed other models. It was hypothesized that the synergized network from the three models can share the strength for broiler chicken feature extraction and further enhance the classification performance. But such a hypothesis requires rigorous verification.

To train those classifiers for custom bird feature extraction, a broiler chicken re-identification dataset was formed. Individual focal broiler chickens were randomly selected from each pen and cropped to form the dataset. A total of 20 focal broiler chickens in each week (weeks 2–7) were picked from each of 48 pens, and 25 consecutive frames were extracted from each broiler chicken, resulting in a total of 5,760 samples and 144,000 image frames for custom training. The focal birds were picked randomly from feeder, drinker, and open areas of each pen. Some sample images were presented in Fig. 5. It should be noted that the drinking line was arranged horizontally, and the vertical drinking line in Fig. 5 was adjusted manually only for the tight layout of the sample images. It should be noted that the sample images in Fig. 5 were extracted directly from the real broiler chicken images instead of data augmentation.

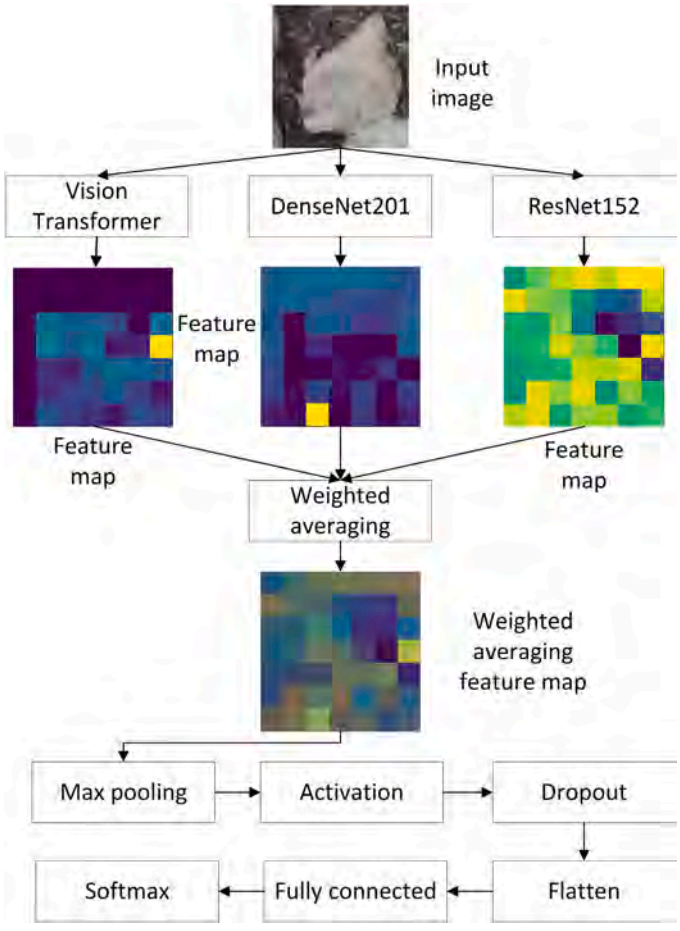

Fig. 4. Overall structure of the proposed deep learning image classification framework.

The input images were set with a fixed random seed number to ensure consistent data learning among various models. The data was augmented with horizontal and vertical flipping and center cropping to increase data variations during training, resized into 224 pixels, and normalized with default matrix that was ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]). The data was split into 80 % for training and 20 % for testing. Models were trained with the stochastic gradient descent optimization algorithm, learning rate of 0.001, momentum of 0.9, step size of 7, gamma of 0.1, weight decay rate of 0.0005, and epoch number of 50. During training, the cross-entropy loss (Eq. (14)) and accuracy (Eq. (15)) curves were depicted by TensorBoard to visualize model performance in real time.

$$Loss = -\frac{1}{N}\left[\sum_{i=1}^{N}[t_i \log(p_i) + (1-t_i)\log(1-p_i)]\right] \tag{14}$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{15}$$

where t_i is the truth value taking a value 0 or 1 and p_i is the Softmax probability for the i^{th} data point out of N data points.

After training, classical feature extraction evaluation metrics were used to evaluate the custom trained feature extractors. Cosine similarity (Eq. (16)) is used to measure the directional similarity between feature vectors of an object across different frames, while Euclidean distance (Eq. (17)) calculates the straight-line distance between those vectors, both being employed for re-identification to determine if an object detected in one frame is the same as one seen in a previous frame. This is of primary importance as tracking algorithms associate the same objects based on the similarity. To calculate the two metrics, the trained feature extractor was first deployed to extract the feature maps from the cropped consecutive frames of the same broiler chickens. Then those feature maps were flattened into feature vector embeddings for further calculation. The values were averaged on a broiler chicken basis. A value closer to 1.0 for cosine similarity represents higher similarity between two feature vectors, while a value closer to 0.0 for Euclidean distance signifies greater similarity.

$$\cos\theta = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{\|\boldsymbol{A}\|\cdot\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} x_i^c x_i^p}{\sqrt{\sum_{i=1}^{n}(x_i^c)^2}\sqrt{\sum_{i=1}^{n}(x_i^p)^2}} \tag{16}$$

$$d(\boldsymbol{A},\boldsymbol{B}) = \sqrt{(\boldsymbol{A}-\boldsymbol{B})(\boldsymbol{A}-\boldsymbol{B})^T} = \sqrt{\sum_{i=1}^{n}(x_i^c - x_i^p)^2} \tag{17}$$

where $\boldsymbol{A}(x_1^c, x_2^c, \ldots, x_i^c)$ represents the feature vector of a feature map for the current frame; while $\boldsymbol{B}(x_1^p, x_2^p, \ldots, x_i^p)$ represents the feature vector of a feature map for the previous consecutive frame.

The Gradient-weighted Class Activation Mapping (Grad-CAM) [46] to enhance the visual explanations of custom trained feature extractors for generating feature maps to associate re-identification in tracking. The Grad-CAM algorithm decoded the optimal deep learning classifier after model comparison and generated heatmaps of the last layer outputs given the input of cropped bird images. The heatmaps highlighted the most important ROI with red color and the least important ROI with

Fig. 5. Sample images in the re-identification dataset.

blue color.

After comprehensive evaluation and optimization, the extracted feature maps generated from the optimal classifier were fed into tracking algorithms to enhance bird re-identification.

2.7. Kinematics-aware machine learning classification

The kinematics-aware machine learning classifier was integrated into tracking algorithm for further reducing ID switch and enhancing the consistency of bird tracking.

Firstly, the developed broiler chicken detector (Section 2.4) was integrated into a tracking algorithm, ByteTrack, to continuously track individual broiler chickens. The pixel coordinates (x, y, w, h) along with ID of each bounding box were extracted and stored for further analysis, where x and y are the centroid coordinates of the bounding box, and w and h are the width and height of the box. A total of 20 focal broiler chickens in each week were randomly selected from feeder, drinker, and open areas of each pen and tracked in videos with 10–20 s durations. Each line of data was clearly marked if it was for the same broiler chickens (class: 1) or different ones (class: 0). A total of 5,760 broiler chicken instances and 32,000 data points (16,000 for 1 and 16,000 for 0) were used for the machine learning model development, with 19,200 for data points for training, 6,400 for validation, and 6,400 for testing. A five-fold cross validation strategy was deployed to develop machine learning classifiers.

Feature extraction involved two types of features from the bounding box information. One was instantaneous features, including raw bounding box information, signal magnitude area, vector magnitude, energy, entropy, body area, angle, perimeter, diagonal length, and diagonal angle. The other was the feature variations between consecutive frames, mainly consisting of the changes of the above features. A total of 26 dimensions or columns of features were extracted. Table 2 shows

Table 2
Formulas for calculating features on machine learning model classification.

Features	Formulas
Raw bounding box information	x_i, y_i, w_i, h_i
Signal magnitude area	$SMA_i = \lvert x_i \rvert + \lvert y_i \rvert$
Vector magnitude	$VM_i = \sqrt{\lvert x_i + y_i \rvert}$
Energy	$NRG_i = \left(x_i^2 + y_i^2\right)^2$
Entropy	$ENT_i = \left(1 + x_i + y_i\right)^2 \times \ln\lvert 1 + x_i + y_i \rvert$
Body area	$BA_i = w_i \times h_i$
Angle	$\theta_i = \tan^{-1}\left(\frac{y_i}{x_i}\right)$
Perimeter	$P_i = 2 \times (w_i + h_i)$
Diagonal length	$DL_i = \sqrt{w_i^2 + h_i^2}$
Diagonal angle	$DA_i = \tan^{-1}\left(\frac{w_i}{h_i}\right)$
Horizontal change	$HC_i = \lvert x_i - x_{i+1} \rvert$
Vertical change	$VC_i = \lvert y_i - y_{i+1} \rvert$
Movement variation	$MV_i = HC_i + VC_i$
Distance change	$DC_i = \left\lvert \sqrt{x_i^2 + y_i^2} - \sqrt{x_{i+1}^2 + y_{i+1}^2} \right\rvert$
Angle change	$AC_i = \lvert \theta_i - \theta_{i+1} \rvert$
Area change	$ARC_i = \lvert BA_i - BA_{i+1} \rvert$
Perimeter change	$PC_i = \lvert P_i - P_{i+1} \rvert$
Diagonal length change	$DLC_i = \lvert DL_i - DL_{i+1} \rvert$
Diagonal angle change	$DAC_i = \lvert DA_i - DA_{i+1} \rvert$
Signal magnitude area change	$SMAC_i = \lvert SMA_i - SMA_{i+1} \rvert$
Vector magnitude change	$VMC_i = \lvert VM_i - VM_{i+1} \rvert$
Energy change	$NRGC_i = \lvert NRG_i - NRG_{i+1} \rvert$
Entropy change	$ENTC_i = \lvert ENT_i - ENT_{i+1} \rvert$

formulas for calculating the two types of features. Each set of features contained a class of 1 or 0 to form the development dataset.

A total of 15 machine learning classification models imported from the Python-based machine learning library (sklearn) are indicated in Table 3. All supervised machine learning models in the sklearn library were assessed for the classification purpose, and only 15 models were compatible and suitable for model development and behavior analysis based on the current data format. There are other machine learning models beyond those selected, but using models from the same library can help to pipeline parameters and hyperparameters and control unnecessary biases/variations (e.g., computing environment differences). The reason for selecting sklearn rather than other libraries is that it is a mature machine learning library and can be easily incorporated into the current model pipeline. The 15 machine learning models included classical models (e.g., linear models) and newer models (e.g., ensemble models). Given the context that most studies for animal behavior classification via triaxial accelerometers investigated 2–5 machine learning models [2,19,45], the 15 models should be sufficient for users to develop and select an optimal one in specific cases. Model training was also based on the Fit function in sklearn.

Table 3
Machine learning classifiers selected for determining the same broiler chickens.

Model name	Functions in sklearn
AdaBoost Classifier	ensemble.AdaBoostClassifier()
Bagging Classifier	ensemble.BaggingClassifier()
Bernoulli Naive Bayes	naive_bayes.BernoulliNB()
Decision Tree Classifier	tree.DecisionTreeClassifier()
Extra Tree Classifier	tree.ExtraTreeClassifier()
Extra Trees Classifier	ensemble.ExtraTreesClassifier()
Gaussian Naive Bayes	naive_bayes.GaussianNB()
Gradient Boosting Classifier	ensemble.GradientBoostingClassifier()
Histogram-based Gradient Boosting Classifier	ensemble.HistGradientBoostingClassifier()
Linear Discriminant Analysis	discriminant_analysis.LinearDiscriminantAnalysis()
Linear Ridge	linear_model.RidgeClassifier()
Quadratic Discriminant Analysis	discriminant_analysis.QuadraticDiscriminantAnalysis()
Random Forest Classifier	ensemble.RandomForestClassifier()
Stochastic Gradient Descent Classifier	linear_model.SGDClassifier()
Support Vector Machine	svm.SVC()

The trained models were evaluated with precision, recall, F1 score, and accuracy as described in Eqs. (9), (10, 11, 15). Based on preliminary observation, the features in Table 2 were in different scales, which could create bias in training and inference and further downgraded the model performance. Thus, normalization of the features based on the mean and standard deviation was conducted to bring all the data points into the same scale (0–1) (Eqs. (18)-(20)).

$$u^j = \frac{\sum_{i=1}^{n} x_i^j}{n} \tag{18}$$

$$\sigma^j = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i^j - u^j\right)^2}{n}} \tag{19}$$

$$z_i^j = \frac{x_i^j - u^j}{\sigma^j} \tag{20}$$

where i represents the i^{th} data point in a row-wise direction; j represents the j^{th} feature in a column-wise direction; n represents the total number of data points; u^j represents the mean of the j^{th} feature; σ^j represents the standard deviation of the j^{th} feature; and z_i^j represents the normalized value of the i^{th} data point for the j^{th} feature.

If every frame was operated with the trained machine learning classifier, it could be computationally expensive. To improve computational efficiency, the classifier was only executed once there was a missing ID between consecutive frames. Detailed implementation and integration procedures of the classifier in a tracking algorithm are depicted in Fig. 6.

2.8. Evaluating tracking performance

The trained pruned broiler chicken detector, feature extractor, and kinematics-aware machine learning classifiers were integrated into six trackers to determine the optimal one, which were ByeTrack [53], StrongSORT [17], DeepSORT [52], Similarity Learning Tracking (SMILEtrack [51]), SparseTrack [38], Robust Association Tracking (BoT-SORT [1]). A total of 20 focal broiler chickens in each week were picked from feeder, drinker, and open areas of each pen for the evaluation, resulting in a total of 5760 bird instances for evaluation. Each broiler chicken was continuously tracked with the developed algorithm in a 30-minute-long video. The tracking performance was evaluated with classical tracking metrics, Multi-Object Tracking Accuracy (MOTA, Eq. (21)) and Multi-Object Tracking Precision (MOTP, Eq. (22)). The longest tracking time (min) was also recorded by manually watching the tracking results of focal broiler chickens before the first ID switches.

$$MOTA = 1 - \frac{\sum_t \left((False\ Negative)_t + (False\ Positive)_t + IDS_t\right)}{\sum_t (Ground\ Truth)_t} \tag{21}$$

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{22}$$

$$d_t^i = \left(\sqrt{\left(x_1^i - x_2^i\right)^2 + \left(y_1^i - y_2^i\right)^2}\right)_t \tag{23}$$

where t represents the time (frame) in a tracking video; i represents the i^{th} bird instance for tracking; IDS represents ID switches; d represents the Euclidean pixel distance between any two tracked centroid coordinates (x_1, y_1) and (x_2, y_2); and c represents total matches made between ground truth and the detection outputs.

With the optimal tracker, tracking performance with different modifications, ages, and ROIs were comparatively evaluated to fully understand the tracking effectiveness. Processing speed (fps) was also

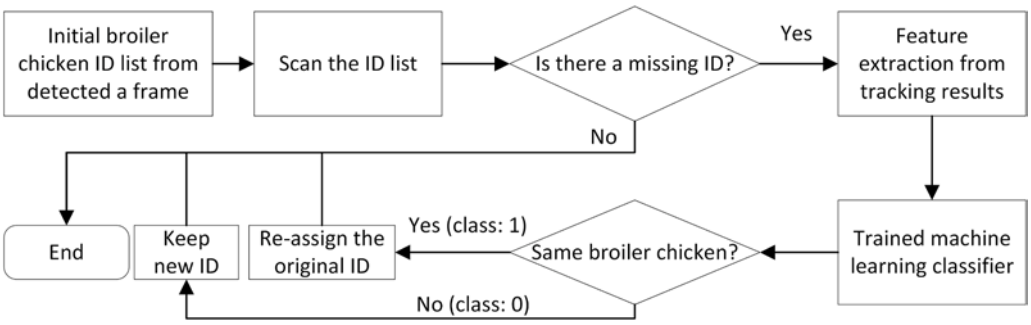

Fig. 6. Integration of the kinematics-aware machine learning classifier to reduce ID switches.

reported. The results were averaged per pen basis and presented as mean ± standard deviation.

3. Results and discussion

3.1. Data preprocessing

The reprojection error (Eq. (6)), a measure of how accurately the calibration model maps 3D points to the 2D image plane, was calculated to assess the calibration quality. The reprojection error for the cameras used in this study was between the range of 0.148 to 0.452 which was less than 0.5 pixels for all of the 48 pens, indicating a high-quality calibration. Example illustrations of the distortion correction effects can be found in Fig. 7. It clearly shows that the edges of the pens were straight out after distortion correction, which can ensure the broiler chickens in far- and near-side were in the same sizes.

Fig. 8 illustrates the deviation percentage before and after camera calibration across ages (in weeks). The deviation percentage increased as the birds aged with 1–4 % in weeks 2 to 4 and 9–12 % in weeks 5 to 7, indicating that camera calibration became more important as broiler chickens grew up. Otherwise, the kinematic features (e.g., velocity or acceleration) calculation could be biased for broiler chickens with different distances from cameras. The error bars represent variability or uncertainty in the measurements. These bars were wider in later weeks, reflecting greater inconsistency in deviation as broiler chickens aged. This could indicate variability in individual broiler chicken sizes and behaviors.

3.2. Performance of detecting broiler chickens

Object models were trained and visualize real-time performance to gain a deep understanding of broiler chicken detection performance. An

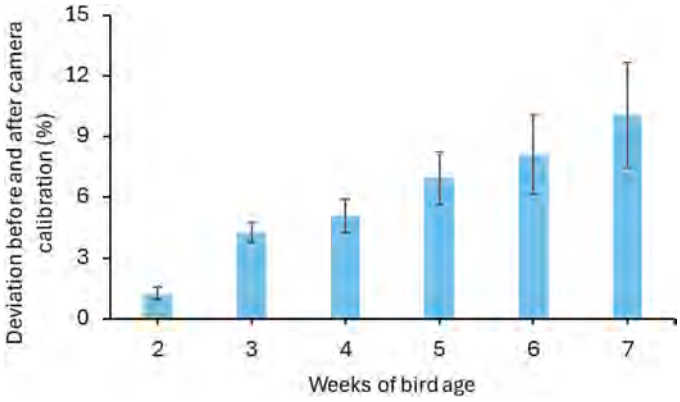

Fig. 8. Deviations before and after camera calibration for broiler chickens at weeks 2 to 7.

example of specific loss metrics of the YOLOv11l model is shown in Fig. 9. The downward trend in train/box_loss indicates improvement in the model's ability to predict accurate bounding boxes over epochs. train/cls_loss reflects how well the model is performing in correctly classifying the broiler chickens. train/dfl_loss represents a direction-focus loss. val/box_loss, val/cls_loss, val/dfl_loss are the corresponding loss metrics during the validation phase. Precision, recall, and mAP are also depicted in Fig. 9. As the epochs progressed, the losses functions decreased steadily while precision, recall, and mAP increased. These trends suggested that the model gained improvement in detecting broiler chickens. The training box loss function is still slightly declining in Fig. 9, but the other loss functions (box loss function for validation and classification and direction-focused loss for both training and

Fig. 7. Example illustrations before (left) and after (right) distortion correction.

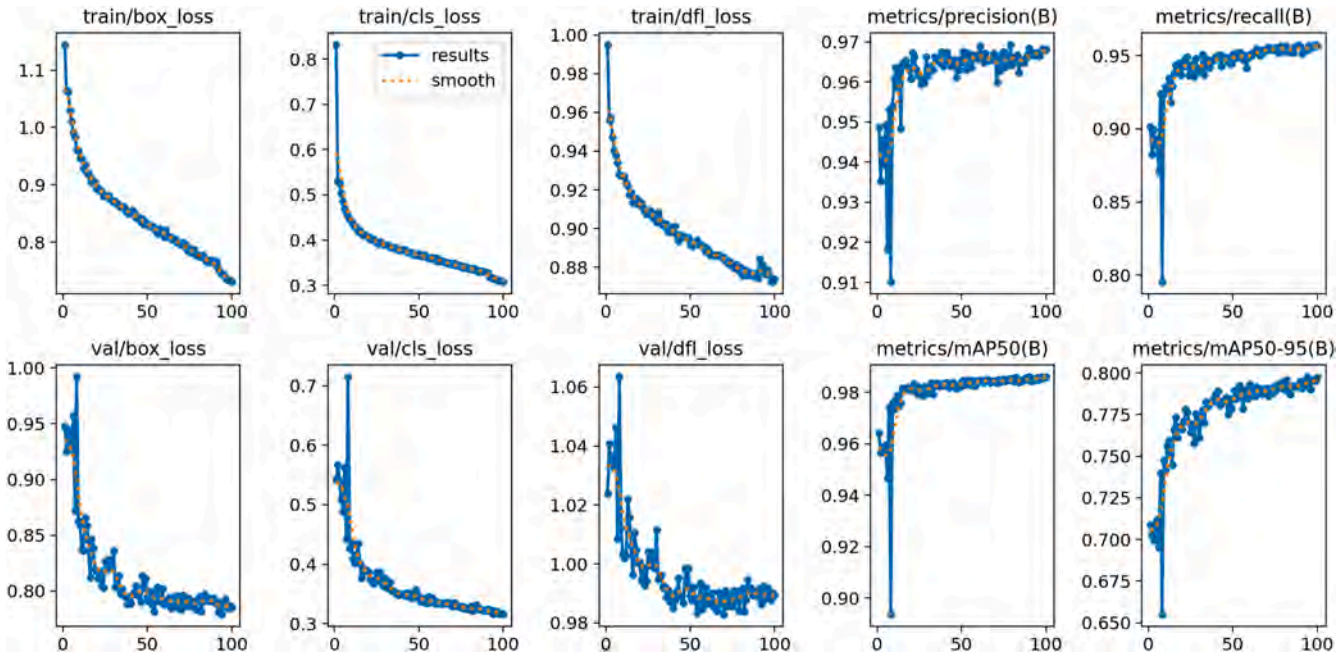

Fig. 9. Training and validation metrics over 100 epochs for the YOLOv11l object detection model. Metrics include box loss, classification (CLS) loss, and direction-focused loss (DFL) for both training and validation phases. Additionally, precision, recall, mean Average Precision (mAP) at IoU=0.50, and mAP over IoU=0.50 to 0.95 are depicted.

validation) and precision and recall curves for training and validation were converged. If the model kept trained over 100 epochs, there might be a risk of overfitting. Incorporating the results from Fig. 9 and Table 4 (with nearly 0.99 mAP50), the models should be well trained for broiler chicken detection.

Table 4 shows the broiler chicken detection performance from ten YOLO object detection models. The range of precision, recall, mAP50, mAP50–95 was 0.966–0.971, 0.938–0.960, 0.981–0.986, and 0.754–0.805, respectively. It took 0.637–1.813 h to fully train a single YOLO model with 100 epochs. The YOLOv8l outperformed other versions of YOLOv8 models with 0.971 precision, 0.960 recall, 0.985 mAP, 0.802 mAP50–95, and training time of 1.731 h. Among all the object detection models tested, the YOLOv11x performed better for broiler chicken detection than others, with 0.968 precision, 0.960 recall, 0.986 mAP50, and 0.805 mAP50–95. Meanwhile, it costed less time (1.217 h) for training the model. Considering all these factors, the YOLOv11x was selected for broiler chicken detector and further optimization in this case.

Some broiler chicken detection examples via the trained YOLOv11x can be found on Fig. 10. The confidence scores of detecting the broiler chickens in weeks 2 and 5 were 0.33–0.89 and 0.77–0.91, respectively. As broiler chickens aged, their body size and features became more distinctive compared to the young ones, thus improving the detection confidence scores.

Table 4
Comparison of various object detection models based on their broiler chicken testing performance metrics.

Model	Precision	Recall	mAP50	mAP50–95	Model training time (h)
YOLOv8n	0.966	0.942	0.982	0.758	1.454
YOLOv8s	0.968	0.951	0.985	0.783	1.438
YOLOv8m	**0.971**	0.952	0.985	0.794	1.595
YOLOv8l	0.969	0.959	**0.986**	0.802	1.731
YOLOv8x	0.967	0.955	0.985	0.801	1.813
YOLOv11n	0.966	0.938	0.981	0.754	**0.637**
YOLOv11s	0.968	0.951	0.984	0.779	0.674
YOLOv11m	0.967	0.954	0.985	0.798	0.814
YOLOv11l	0.968	0.958	**0.986**	0.802	1.043
YOLOv11x	0.967	**0.960**	**0.986**	**0.805**	1.217

Note: YOLO = You Only Look Once; mAP50 = mean Average Precision with the threshold of Intersection over Union (IoU) over 0.5; mAP50–95 = mean Average Precision with the threshold of IoU in between 0.50 and 0.95. Bold fonts indicate the best performance.

Deep learning-based computer vision algorithms have been evolving and progressing to advance automated behavior monitoring in broiler chickens. Nasiri et al. [41] developed a video recognition-based pipeline using 3D CNN and YOLO model to identify stretching and preening in broiler chickens and achieved 96.7 % accuracy. Li et al. [32] developed a hen behavior detectors using a Mask R-CNN to identify preening in hens and achieved 94.2 % accuracy. Guo et al. [21] developed a network model using CNN and DenseNet-264 to identify general behaviors (i.e., feeding, drinking, standing, and resting) in broilers and achieved 97 % accuracy. Li et al. [33] developed image processing algorithms using CNN-based and YOLO model to identify feeding and drinking behavior in broiler chickens and achieved >89 % accuracy, and Chen et al. [10] developed an automatic detection system using YOLOv7-tiny and SORT [6] algorithm to identify dispersion and movement in broiler chickens and achieved 98.2 % precision and 95.3 % tracking accuracy. Compared to the previous studies, the current study presented the highest precision of 0.971, recall of 0.960, mAP50 of 0.986, and mAP50–95 of 0.805, suggesting robust broiler chicken detection performance.

3.3. Pruning performance of object detection models for broiler chicken detection

The selected YOLOv11x model was further pruned to enhance bird prediction efficiency (Table 5). The pruning ratio ranged from 0.01 to 0.15. The precision (0.966–0.967), recall (0.955–0.957), mAP50 (0.985–0.986), and mAP50–95 (0.799) were similar within the range of 0.01–0.09. However, when the pruning ratio increased from 0.09 to 0.10, the broiler chicken detection performance dropped sharply, with precision from 0.966 to 0.467, recall from 0.957 to 0.786, mAP50 from 0.986 to 0.576, and mAP50–95 from 0.799 to 0.403. The performance remained stably low between 0.10 to 0.15. A pruning rate of 0.09 was selected for the model. After pruning, the model was further fine-tuned, resulting in 0.956 recall, 0.968 precision, and 0.986 mAP50. These

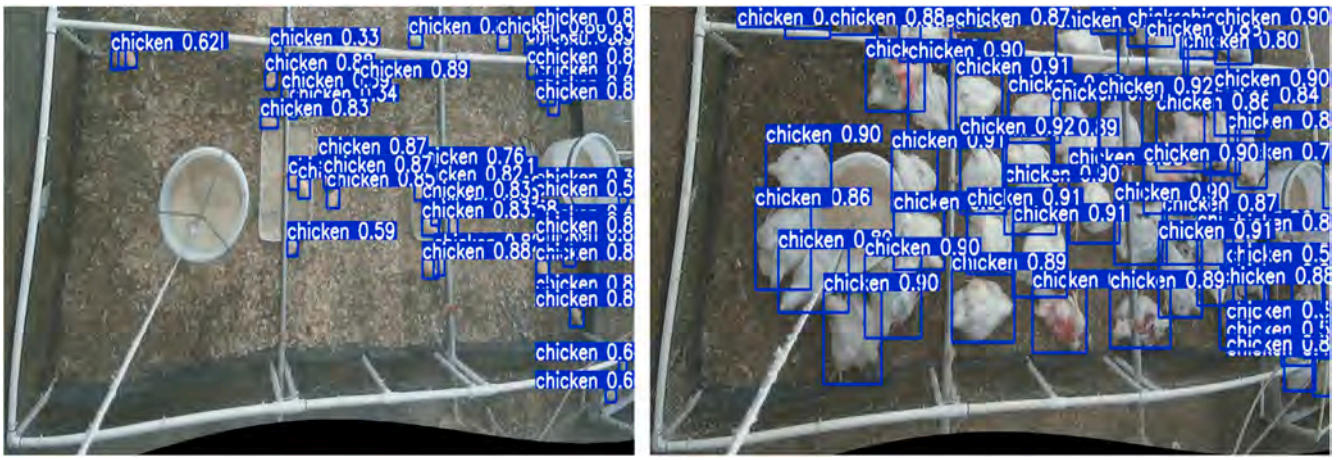

Fig. 10. Example illustrations of broiler chicken detection in weeks 1 (left) and 5 (right) via YOLOv11x.

Table 5
The performance metrics of pruning the object detection model (YOLOv11x) at various proportion.

Proportion	Precision	Recall	mAP50	mAP50–95
0.01	0.967	0.955	0.985	0.799
0.02	0.967	0.955	0.985	0.799
0.03	0.967	0.955	0.985	0.799
0.04	0.967	0.956	0.985	0.799
0.05	0.967	0.956	0.985	0.799
0.06	0.967	0.955	0.985	0.799
0.07	0.967	0.955	0.985	0.799
0.08	0.966	0.956	0.985	0.799
0.09	**0.966**	**0.957**	**0.986**	**0.799**
0.10	0.467	0.786	0.576	0.403
0.11	0.467	0.786	0.576	0.403
0.12	0.467	0.786	0.576	0.403
0.13	0.467	0.786	0.576	0.403
0.14	0.467	0.786	0.576	0.403
0.15	0.467	0.786	0.576	0.403

Note: YOLO = You Only Look Once; mAP50 = mean Average Precision with the threshold of Intersection over Union (IoU) over 0.5; mAP50–95 = mean Average Precision with the threshold of IoU in between 0.50 and 0.95.

results indicate only slight changes compared to the original model.

Fig. 11 shows the relationship between processing speed and precision for various pruning amounts applied to the YOLOv11x model. The sweet spot was identified at 0.09 pruning amount, with the precision of 0.966 and processing speed of 60 fps. With the pruning, the YOLOv11x reduced the precision by 0.01 but increased the processing speed from 46.5 fps to 60 fps. The selection of the sweet spot reflected a conscious decision to balance model efficiency (through pruning for faster processing) while maintaining high detection performance (precision).

3.4. Performance of broiler chicken re-identification with custom trained feature extractors

Fig. 12 shows the accuracy and loss for training and validation for the proposed deep learning image classifier (Fig. 4). The model learned patterns effectively from the feed-in data. Meanwhile, the performance gap between training and validation was small, which were less than 0.20 for loss and less than 0.15 for accuracy at the end of the training. This result demonstrated that the model was trained well while could generalize for unseen data during validation. Because of small gaps between training and validation losses, the proposed network was not overfitted. Other image classifiers listed in Section 2.6 followed similar patterns during training. If the performance gaps between training and validation were significant (i.e., >0.3), early stop was implemented and hyperparameters were adjusted to ensure the training and validation had a small gap to avoid overfitting.

Table 6 presents a comparison of various deep learning models based on two performance metrics: cosine similarity and Euclidean distance, reflecting the quality of feature representations of re-identification for object tracking. Higher cosine similarity values and lower Euclidean distance values indicate better alignment of model predictions with the ground truth in the feature space. The DenseNet201, ResNet152, and Vision Transformer had higher cosine similarity values (0.918–0.932) and lower Euclidean distance values (0.034–0.113) than other classifiers due to their superior architecture (e.g., transformer-based architecture)

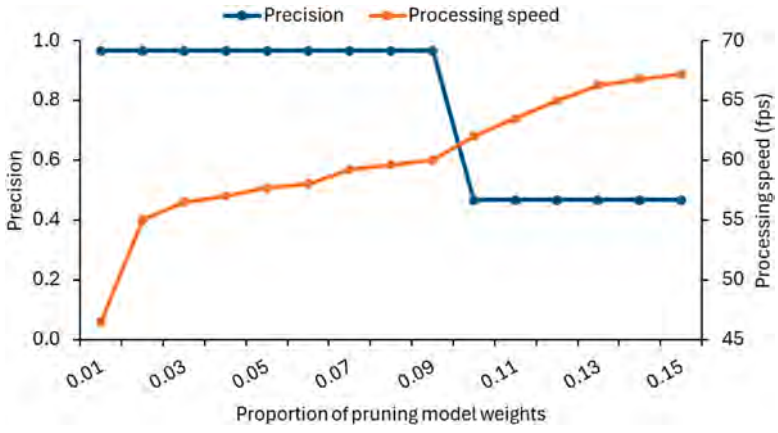

Fig. 11. Precision and processing speed with different proportions of pruning model weights for YOLOv11x.

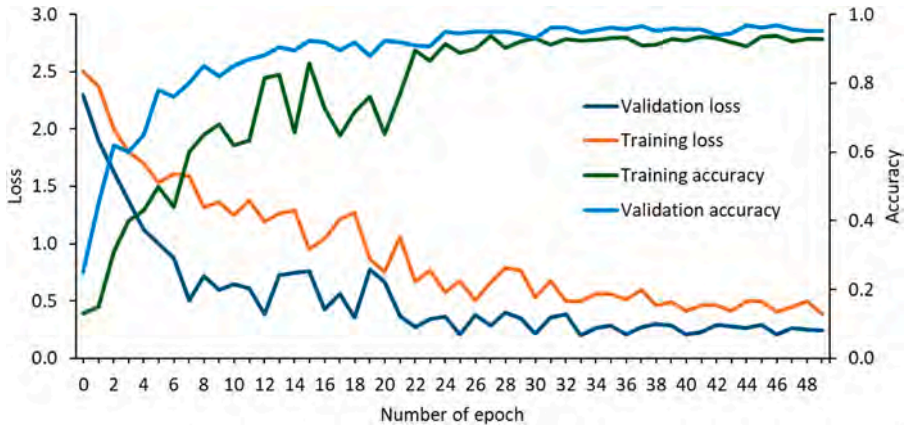

Fig. 12. Training and validation performance of the proposed deep learning image classifier.

Table 6
Cosine similarity and Euclidean distance metrics for evaluating re-identification performance of custom-trained deep learning image classifiers.

Model	Cosine similarity	Euclidean distance
DenseNet201	0.918±0.027	0.084±0.026
EfficientNetB7	0.600±0.056	0.929±0.165
InceptionResNetV2	0.905±0.032	0.159±0.081
MobileNetV3Large	0.701±0.042	0.774±0.187
ResNet152	0.920±0.028	0.113±0.048
Vision Transformer	0.932±0.027	0.034±0.015
VGG19	0.842±0.026	0.504±0.207
Xception	0.890±0.027	0.236±0.097
Proposed method	0.956±0.032	0.020±0.007

and advanced connection schemes (e.g., shortcut connections) [16,23,26]. The three models were merged into a new model (Fig. 4). Such a combination can ensure that the advantageous components from the respective models can be complemented and further enhance the classification performance. The proposed method presented better performance in the two metrics (0.956±0.032 for cosine similarity and 0.020 ±0.007 for Euclidean distance) and showed superiority in feature extraction, thus being selected to integrated into the tracking framework.

With the proposed method, we run the Grad-CAM for generating the heatmaps (Fig. 13) of broiler chickens providing visual explanations of which regions in images contribute most to the model's decision-making. The proposed method had a large variation of attentions during the decision-making process. The top three heatmaps in Fig. 13 show most of the attention concentrated in a narrow horizontal band, which was the area of drinking line included in the training dataset (Fig. 5). The bottom three heatmaps in Fig. 13 show the attentions focusing on the broiler chickens' body parts effectively. Those results suggest that the facilities (e.g., feeder or drinker) interacting with broiler chickens may enhance feature extractions and further enhance the consistency of tracking.

3.5. Kinematics-aware machine learning classification performance

With the input features in Table 2, a total of 15 machine learning classifiers were trained with the outputs of the same broiler chickens (class: 1) and different ones (class: 0). Before normalization, the accuracy, precision, recall, and F1 score were 0.618–0.610, 0.597–0.610, 0.800–0.803, and 0.795–0.782, respectively (Table 7). After normalization, the four metrics were 0.715–0.917, 0.715–00,958, 0.718–0.920, and 0.726–0.937, respectively (Table 8). The classification performance was increased 0.073–0.189 with an average of 0.131. The results suggest the necessity of data normalization to bring all features into the same scale for enhanced classification.

Before normalization, the Random Forest Classifier and Extra Trees Classifier performed better than the other 13 classifiers. After normalization, the Extra Trees Classifier outperformed the other 14 classifiers with the highest accuracy of 0.917, precision of 0.958, recall of 0.920, and F1 score of 0.939. The Extra Trees Classifier presented more stable and accurate predictions in determining the same/different broiler chickens, thus being selected to integrated into the tracking framework to enhance multiple broiler chicken tracking.

3.6. Tracking performance

The six tracking algorithms (i.e., StrongSORT, DeepSORT, SMILEtrack, SparseTrack, BoT-SORT, and ByteTrack) were integrated with the YOLOv11x (with 0.09 proportion of weights pruned), proposed deep learning image classifier for custom feature extraction (combining DenseNet201, ResNet152, and Vision Transformer), and Extra Trees Classifier for kinematics-aware classification. Their tracking performance of multiple broiler chickens from weeks 2 to 7 is presented in Table 9. StrongSORT, DeepSORT, and SMILEtrack performed well in accuracy and precision but were significantly slower than ByteTrack. SparseTrack and BoT-SORT performed poorly in accuracy, precision, and speed, making them less competitive compared to other tracking algorithms. In sum, the ByteTrack was determined as the optimal tracking algorithm in this study with the highest MOTA (0.904±0.073), MOTP (0.953±0.057), and processing speed (30.1±3.3 fps) and longest tracking duration (17.3 min).

Additional ablation studies were conducted for the modified ByteTrack algorithm to understand the role of each component in decision-making of multiple broiler chicken tracking (Table 9). Among the three major modifications that were model pruning, feature extraction customization, and kinematics-aware machine learning classification, the feature extraction customization improved the tracking accuracy the most with the increment of 0.146 MOTA and 0.163 MOTP while removing the kinematics-aware machine learning classifier could increase the processing speed up to 42.3 fps. Both components were critical for consistently tracking individual broiler chickens as removing either of them could downgrade the longest tracking duration to as low as 6.8 min. The ByteTrack without object detection model pruning can maintain the highest MOTA (0.912±0.063) and MOTP (0.958±0.032) but had reduced speed (24.7±3.5 fps). The longest tracking duration with or without model pruning was not considerably different. For balancing the accuracy and efficiency, the pruned broiler chicken detector with 0.09 proportion of pruned weights needed to be integrated into the ByteTrack.

The tracking performance of the modified ByteTrack algorithm was broken down into weeks 2 to 7 and presented in Table 10. The tracking performance increased steadily as broiler chicken age increased, with

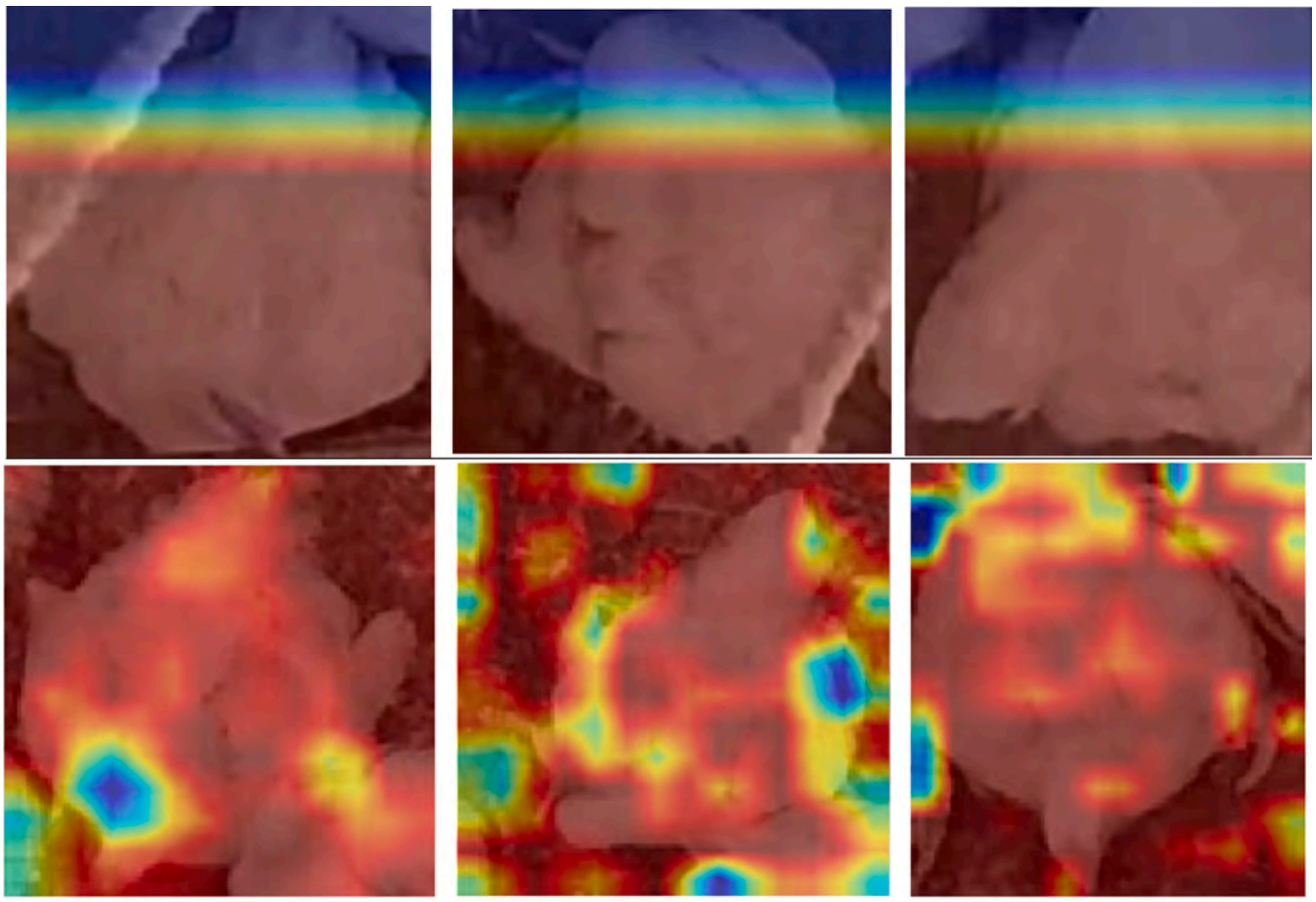

Fig. 13. Heatmaps of the last layer outputs for the proposed deep learning image classifier. The top three and bottom three are from the same broiler chickens, respectively.

Table 7
Machine learning classification performance (without data normalization) in determining the same/different broiler chickens.

Classifier	Accuracy	Precision	Recall	F1-score
AdaBoost Classifier	0.695	0.702	0.672	0.687
Bagging Classifier	0.782	0.763	0.795	0.779
Bernoulli Naive Bayes	0.618	0.642	0.597	0.619
Decision Tree Classifier	0.702	0.717	0.710	0.713
Extra Tree Classifier	0.670	0.665	0.675	0.670
Extra Trees Classifier	0.793	0.795	**0.770**	**0.782**
Gaussian Naive Bayes	0.625	0.610	0.636	0.623
Gradient Boosting Classifier	0.745	0.736	0.741	0.738
Histogram-based Gradient Boosting Classifier	0.754	0.783	0.775	0.779
Linear Discriminant Analysis	0.650	0.672	0.667	0.669
Linear Ridge	0.694	0.688	0.696	0.692
Quadratic Discriminant Analysis	0.636	0.621	0.600	0.610
Random Forest Classifier	**0.800**	**0.803**	0.760	0.781
Stochastic Gradient Descent Classifier	0.641	0.645	0.651	0.648
Support Vector Machine	0.678	0.661	0.646	0.653
Average	0.699	0.700	0.693	0.696

Table 8
Machine learning classification performance (with data normalization) in determining the same/different broiler chickens.

Classifier	Accuracy	Precision	Recall	F1-score
AdaBoost Classifier	0.829	0.839	0.830	0.834
Bagging Classifier	0.896	0.919	0.909	0.914
Bernoulli Naive Bayes	0.715	0.715	0.737	0.726
Decision Tree Classifier	0.852	0.822	0.825	0.823
Extra Tree Classifier	0.806	0.816	0.828	0.822
Extra Trees Classifier	**0.917**	**0.958**	**0.920**	**0.939**
Gaussian Naive Bayes	0.777	0.768	0.742	0.755
Gradient Boosting Classifier	0.840	0.874	0.842	0.858
Histogram-based Gradient Boosting Classifier	0.914	0.884	0.911	0.897
Linear Discriminant Analysis	0.812	0.801	0.800	0.800
Linear Ridge	0.798	0.796	0.816	0.806
Quadratic Discriminant Analysis	0.738	0.750	0.718	0.734
Random Forest Classifier	0.902	0.922	0.913	0.917
Stochastic Gradient Descent Classifier	0.762	0.790	0.757	0.773
Support Vector Machine	0.831	0.812	0.835	0.823
Average	0.826	0.831	0.826	0.828

the highest MOTA at week 7 (0.917±0.041), MOTP at week 6 (0.998 ±0.007), and longest tracking duration at week 5 (17.3 min). MOTP remained high for all ages (0.982–0.998), indicating that the algorithm can accurately localize birds when successfully tracked. As broiler chickens grew up, their physical features became more distinct with more feature coverage and sharper background contrast, improving the algorithm's ability to detect and track them accurately. In addition, older broiler chickens were lazier in moving and exhibit more predictable movement patterns, making tracking easier.

The tracking performance via the modified ByteTrack was also summarized into three major regions in a pen, which were feeder, drinker, and open areas (Table 11). The tracking performance was

Table 9
Multiple broiler chicken tracking performance of various tracking algorithms.

Tracking algorithm	MOTA	MOTP	Processing speed (fps)	Longest tracking duration (min)
Modified StrongSORT	0.827 ±0.060	0.946 ±0.040	7.5±3.7	10.9
Modified DeepSORT	0.850 ±0.053	0.932 ±0.064	10.9±2.9	12.1
Modified SMILEtrack	0.888 ±0.020	0.949 ±0.035	3.7±1.7	14.3
Modified SparseTrack	0.868 ±0.029	0.902 ±0.064	18.6±2.3	12.6
Modified BoT-SORT	0.795 ±0.032	0.884 ±0.026	5.2±3.4	8.8
Modified ByteTrack	0.904 ±0.073	0.953 ±0.057	30.1±3.3	**17.3**
Modified ByteTrack without the pruned object detector	**0.912 ±0.063**	**0.958 ±0.032**	24.7±3.5	17.0
Modified ByteTrack without the custom feature extractor	0.758 ±0.031	0.790 ±0.026	33.5±3.1	7.2
Modified ByteTrack without the kinematics-aware machine learning classifier	0.850 ±0.076	0.868 ±0.054	**38.1±4.2**	6.8

Note: SORT = Simple Online and Realtime Tracking; MOTA = Multi-Object Tracking Accuracy; MOTP = Multi-Object Tracking Precision.

Table 10
Tracking performance of different broiler chicken age groups using the modified ByteTrack.

Bird age (weeks)	MOTA	MOTP	Longest tracking duration (min)
2	0.865±0.048	0.982±0.014	10.5
3	0.853±0.037	0.985±0.008	11.2
4	0.872±0.060	0.990±0.011	11.8
5	0.897±0.045	0.981±0.009	**17.3**
6	0.893±0.032	**0.998 ±0.007**	16.5
7	**0.917 ±0.041**	0.993±0.012	17.2

Note: MOTA = Multi-Object Tracking Accuracy; MOTP = Multi-Object Tracking Precision.

Table 11
Tracking performance of different regions using the modified ByteTrack.

Area	MOTA	MOTP	Longest tracking duration (min)
Feeder	0.828±0.129	0.982±0.019	11.6
Drinker	0.891±0.085	0.981±0.015	14.6
Open area	**0.907±0.041**	**0.993±0.011**	**17.3**

Note: MOTA = Multi-Object Tracking Accuracy; MOTP = Multi-Object Tracking Precision.

downgraded around the necessary resources (e.g., feeders and drinkers), with the MOTA of 0.828–0.891, MOTP of 0.982–0.981, and longest tracking duration of 11.6–14.6 min. The best tracking performance was observed in open area with the longest tracking duration of 17.1 min. Feeder and drinker due to necessity were prone to crowing, leading to partial or complete occlusion of broiler chickens, making tracking difficult. Complete social interactions around feeders and drinkers such as competition and avoidance may result in erratic movements or overlapping, complicating accurate tracking.

3.7. Comparison with previous investigations

Based on the comparison with the previous studies (Table 12), the proposed method presented superior performance of tracking multiple individual broiler chickens in group settings. The set of algorithms increased at least 4 min by tracking multiple individual broiler chickens. It should be noted that the results of previous studies listed in Table 12 were conducted in less challenging environments than the current study. They either contained a short time of tracking (e.g., within several days), loose densities, or single pens. In comparison, this study tracked broiler chickens from high density (37 birds in a 3.6 m^2 pen), multiple ages and sizes (weeks 2 to 7), changing environments (e.g., light intensity (5–30 lx), dust and moisture levels (dry in the beginning and wet in the later phase), or litter conditions (fresh litter and caked litter)), and practical regions (feeder, drinker, and open areas). Those variations helped to develop and evaluate generalizable trackers in practical settings for broiler chickens.

3.8. Limitations and future directions

A camera was installed and adjusted to cover most areas of a pen, but the installation configurations (e.g., tilted angles, installation places relative to a pen) could be varied with pens. Therefore, a total of 48 sets of calibration parameters were stored and implemented for corresponding pens for distortion correction. However, the calibration results were not optimal, which can be reflected in Figs. 7 and 10. During calibration, a technician handled the checkerboard and randomly placed it in different places of pen. Such operations could introduce inconsistency of matrix transformation, resulting in variations for the distortion correction results. A consistent data acquisition framework like aluminum framework is recommended to accommodate the checkerboard to enhance matrix consistency and improve distortion correction results.

The weight-based pruning method was built based on the PyTorch existing function with the key parameter, k, to determine proportion of weights pruned (instead of number of parameters) for enhancing the efficiency of the tracking framework. Number of parameters were not presented in the existing function. With that major purpose, we primarily tested the accuracy and processing speed of broiler chicken detection regarding different proportions of weights as indicated in Table 5 and Fig. 11. Diverse methods of neuron-level pruning, layer-level pruning, or parameter-level pruning should be investigated in the future to understand the effects of different pruning components more precisely.

Although the proposed modeling system presented great potential in enhancing consistency and efficiency of multiple broiler chicken tracking, the system heavily relied on post-hoc analysis of videos, such as the kinematic-aware machine learning classifiers. That is not

Table 12
Tracking performance in different studies.

Author	Algorithm	Broiler chicken age (weeks)	Longest tracking duration (min)
Fang et al. [18]	AlexNet with OpenCV trackers	NA	9.5
Doornweerd et al. [15]	YOLOv7-tiny with SORT	3	12.4
Merenda et al. [40]	StrongSORT with OSNet and machine learning classifier	4	3.5
This study	ByteTrack with the pruned YOLOv11m, custom feature extractor, and kinematics-aware machine learning classifier	2–7	17.3

Note: SORT = Simple Online and Realtime Tracking; YOLO = You Only Look Once; NA = Not Applicable.

supportive of real-time bird tracking. More generalizable and efficient modeling systems need to be researched in the future to achieve real-time tracking. To scale the tracking algorithm from the controlled experimental setup to a commercial farm with larger flocks (>10,000 birds), a ceiling-mounted robot [13] should be installed at the similar height of the current camera system. These ceiling-mounted robots should also be stationary for a certain duration (e.g., half an hour) in a region to capture the dynamic variations and environments (e.g., moving equipment, varying stocking densities). Integrating additional modalities, such as audio [3], thermography [9], and depth [28], may be helpful for further boosting tracking performance. In addition, challenges (e.g., disease and stress) experiments need to be conducted, and their corresponding kinematics including movement speed, acceleration, and trajectories have to be extracted with the trackers. That information will aid in producing better genetics, nutrition, and manage strategies to against the challenges.

The tracking framework was optimized with custom training of object detection models, strengthening re-identification network for broiler chicken appearance feature extraction, and kinematics-aware machine learning classification. The longest tracking duration (17.3 min) outperformed the previous research but is still limited to commercial monitoring needs in hourly or daily basics. More methods from MOT in crowed pedestrians should be investigated to improve the tracking performance, reduce the accumulated errors, and recover ID switches. Li et al. [34] advanced MOT tracking algorithm through occlusion-awareness and trajectory optimization. Cai et al. [7] proposed instance-aware MOT tracking schemes with motion consistency. Lv et al. [39] deviced a visibility branch to predict the object occlusion level, and a predicted visibility map was used in both feature refinement model and visibility-guided two-stage association strategy. Those are proven networks in human areas and should be carefully integrated and evaluated for multiple broiler chicken tracking.

This study was primarily set up with practical farming configurations for advancing tracking algorithm development. Broiler chickens were raised in full brightness (20–30 lx) from days 0–9 for brooding purposes and in dim environments (~5 lx) from days 10–49 for grow-out. Fresh and reused litter were distributed evenly for the 48 pens. The temperature was set at a decreasing manner (33.9–21.1 °C) from days 0–49. The stocking density can be as high as lower than 1 ft^2 per bird in week 7. Occlusion scenarios (e.g., overlapping birds, facility obstructions) were included to challenge the model as suggested in Fig. 10. The occlusion in this study was even more challenging than commercial farms as birds in commercial farms can have more space to move. To fit the practical settings, images were evenly extracted from each of weeks 2 to 7 and 48 pens to include the variations of occlusion, litter conditions, temperature, humidity, lighting intensities, body sizes, and feather coverage rates. Despite comprehensive setup, most of the factors were mixed together making it difficult to quantify certain factors on tracking performance. For instance, as bird aged from weeks 2 to 7 (Table 10), the MOTA and MOTP increased due to mixing factors of less occlusions, clearer body boundaries, and sharper contrast between litter floor and broiler chickens instead of single factors such as occlusion only. Future works are recommended to study those variations independently and collaboratively.

4. Conclusion

This article aimed to enhance the consistency and efficiency of tracking multiple individual broiler chickens in group settings. The YOLOv11x outperformed other nine object detection models and achieved over 0.96 precision, recall, and mAP50 for detecting broilers from weeks 2 to 7. The YOLOv11x with 0.09 proportion of pruned weights can increase 13.5 fps processing speed while not compromising broiler chicken detection performance, compared to the YOLOv11x with full custom weights. The proposed feature extractor combined with Vision Transformer, ResNet152, and DenseNet201 models outperformed other eight deep learning image classifier and enhanced re-identification performance via increasing the MOTA and MOTP by nearly 0.15. The kinematics-aware machine learning classifier (integrated with the extra trees classifier out of 15 compared machine learning classifiers) can improve 0.05–0.09 MOTA and MOTP. The modified ByteTrack performed better in MOTA, MOTP, processing speed, and longest tracking duration than the other five trackers. By putting all components together, the integrated model system achieved 0.904±0.073 MOTA, 0.953±0.057, MOTP, 30.1±3.3 fps processing speed, and 17.3 min longest tracking duration for tracking multiple individual broiler chickens from weeks 2 to 7 and in feeder, drinker, and open areas.

Ethics statement

Not applicable: This manuscript does not include human or animal research.

If this manuscript involves research on animals or humans, it is imperative to disclose all approval details.

If Yes, please provide your text here:

All procedures were approved by Institutional Animal Care and Use Committees at the University of Georgia (protocol number: A2023 07–016-Y1-A0).

Statement

During the preparation of this work the authors used Microsoft Copilot in order to improve the readability and language of the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the content of the published article.

CRediT authorship contribution statement

Guoming Li: Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Sai Akshitha Reddy Kota:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Tongshuai Liu:** Writing – review & editing, Data curation. **Oluwadamilola Moyin Oso:** Writing – review & editing, Data curation. **Venkat Umesh Chandra Bodempudi:** Writing – review & editing, Data curation. **Mahtab Saeidifar:** Writing – review & editing, Data curation. **Ehsan Asali:** Writing – review & editing, Data curation. **Aravind Mandiga:** Writing – review & editing, Data curation. **Jin Lu:** Writing – review & editing. **Geng Yuan:** Writing – review & editing. **Ahmad Banakar:** Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This research was supported by the Cobb Research Initiative.

Data availability

Data will be made available on request.

References

[1] N. Aharon, R. Orfaig, B.-Z. Bobrovsky, BoT-SORT: robust associations multi-pedestrian tracking, arXiv preprint (2022), https://doi.org/10.48550/arXiv.2206.14651.

[2] P. Balasso, G. Marchesini, N. Ughelini, L. Serva, I. Andrighetto, Machine learning to detect posture and behavior in dairy cows: information from an accelerometer on the animal's left flank, Animals 11 (2021), https://doi.org/10.3390/ani11102972.
[3] A. Banakar, M. Sharafi, G. Li, Relationship between reproductive indicators and sound structure in broiler breeder roosters, Appl. Anim. Behav. Sci. 270 (2024) 106135, https://doi.org/10.1016/j.applanim.2023.106135.
[4] N. Ben Sassi, X. Averós, I. Estevez, Technology and poultry welfare, Animals 6 (2016), https://doi.org/10.3390/ani6100062.
[5] D. Berckmans, General introduction to precision livestock farming, Anim. Front. 7 (2017) 6–11, https://doi.org/10.2527/af.2017.0102.
[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3464–3468.
[7] J. Cai, Y. Wang, H. Zhang, H.-M. Hsu, C. Ma, J.-N. Hwang, Ia-mot: instance-aware multi-object tracking with motion consistency, arXiv preprint arXiv 2006 (2020) 13458.
[8] M. Campbell, P. Miller, K. Díaz-Chito, X. Hong, N. McLaughlin, F. Parvinzamir, J. Martínez Del Rincón, N. O'Connell, A computer vision approach to monitor activity in commercial broiler chickens using trajectory-based clustering analysis, Comput. Electron. Agric. 217 (2024) 108591, https://doi.org/10.1016/j.compag.2023.108591.
[9] Z. Cao, Z. Shi, X. An, G. Li, Evaluating the thermal insulation of dairy barns in cold regions via infrared thermography, in: Animal Environment and Welfare–Proceedings of International Symposium, 2017, pp. 53–60.
[10] B.-L. Chen, T.-H. Cheng, Y.-C. Huang, Y.-L. Hsieh, H.-C. Hsu, C.-Y. Lu, M.-H. Huang, S.-Y. Nien, Y.-F. Kuo, Developing an automatic warning system for anomalous chicken dispersion and movement using deep learning and machine learning, Poult. Sci. 102 (2023) 103040.
[11] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[12] Cobb, Cobb500 Broiler: performance & nutrition supplement, Accessed on December 2024. Aviable at, https://www.cobbgenetics.com/assets/Cobb-Files/2022-Cobb500-Broiler-Performance-Nutrition-Supplement.pdf, 2022.
[13] G. da Rocha Balthazar, R.M.F. Silveira, I.J.O. da Silva, Use of robotics in broiler production systems: a relationship between technology, environment and production, Trop. Anim. Health Prod. 57 (2025) 142, https://doi.org/10.1007/s11250-025-04381-z.
[14] J. Deng, W. Dong, R. Socher, L.J. Li, L. Kai, F.-F. Li, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[15] J.E. Doornweerd, R.F. Veerkamp, B. de Klerk, M. van der Sluis, A.C. Bouwman, E. D. Ellen, G. Kootstra, Tracking individual broilers on video in terms of time and distance, Poult. Sci. 103 (2024) 103185, https://doi.org/10.1016/j.psj.2023.103185.
[16] A. Dosovitskiy, An image is worth 16×16 words: transformers for image recognition at scale, arXiv preprint arXiv 2010 (2020) 11929.
[17] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, H. Meng, StrongSORT: make DeepSORT great again, IEEE Trans. Multimed. 25 (2023) 8725–8737, https://doi.org/10.1109/TMM.2023.3240881.
[18] C. Fang, J. Huang, K. Cuan, X. Zhuang, T. Zhang, Comparative study on poultry target tracking algorithms based on a deep regression network, Biosys. Eng. 190 (2020) 176–183, https://doi.org/10.1016/j.biosystemseng.2019.12.002.
[19] B. Fida, I. Bernabucci, D. Bibbo, S. Conforto, M. Schmid, Varying behavior of different window sizes on the classification of static and dynamic physical activities from a single accelerometer, Med. Eng. Phys. 37 (2015) 705–711, https://doi.org/10.1016/j.medengphy.2015.04.005.
[20] J. Frankle, M. Carbin, The lottery ticket hypothesis: finding sparse, trainable neural networks, arXiv preprint arXiv 1803 (2018) 03635.
[21] Y. Guo, S.E. Aggrey, P. Wang, A. Oladeinde, L. Chai, Monitoring behaviors of broiler chickens at different ages with deep learning, Animals 12 (2022) 3390.
[22] R. Hartley, S.B. Kang, Parameter-free radial distortion correction with Center of Distortion estimation, ITPAM 29 (2007) 1309–1321, https://doi.org/10.1109/TPAMI.2007.1147.
[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[24] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, A. Peste, Sparsity in deep learning: pruning and growth for efficient inference and training in neural networks, J. Mach. Learn. Res. 22 (2021) 1–124.
[25] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.
[26] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[27] T. Lei, G. Li, C. Luo, L. Zhang, L. Liu, R.S. Gates, An informative planning-based multi-layer robot navigation system as applied in a poultry barn, Intell. Robot. 2 (2022) 313–332, https://doi.org/10.20517/ir.2022.18.
[28] G. Li, R.S. Gates, M.M. Meyer, E.A. Bobeck, Tracking and characterizing spatiotemporal and three-dimensional locomotive behaviors of individual broilers in the three-point gait-scoring system, Animals 13 (2023), https://doi.org/10.3390/ani13040717.
[29] G. Li, Y. Huang, Z. Chen, G.D. Chesser, J.L. Purswell, J. Linhoss, Y. Zhao, Practices and applications of convolutional neural network-based computer vision systems in animal farming: a review, Sensors 21 (2021), https://doi.org/10.3390/s21041492.
[30] G. Li, Y. Huang, Z. Chen, G.D. Chesser Jr, J.L. Purswell, J. Linhoss, Y. Zhao, Practices and applications of convolutional neural network-based computer vision systems in animal farming: a review, Sensors 21 (2021) 1492.
[31] G. Li, X. Hui, Z. Chen, G.D. Chesser, Y. Zhao, Development and evaluation of a method to detect broilers continuously walking around feeder as an indication of restricted feeding behaviors, Comput. Electron. Agric. 181 (2021) 105982, https://doi.org/10.1016/j.compag.2020.105982.
[32] G. Li, X. Hui, F. Lin, Y. Zhao, Developing and evaluating poultry preening behavior detectors via mask region-based convolutional neural network, Animals 10 (2020) 1762.
[33] G. Li, Y. Zhao, J.L. Purswell, Q. Du, G.D. Chesser Jr, J.W. Lowe, Analysis of feeding and drinking behaviors of group-reared broilers via image processing, Comput. Electron. Agric. 175 (2020) 105596.
[34] Li, M., Zhang, Y., Jia, Y., Yang, Y., 2025. Advancing multi-object tracking through occlusion-awareness and trajectory optimization. Knowledge-based Systems 310, 112930. doi:10.1016/j.knosys.2024.112930.
[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.
[36] G. Liu, H. Guo, A. Ruchay, A. Pezzuolo, Recent advancements in precision livestock farming, Agriculture 13 (2023), https://doi.org/10.3390/agriculture13091652.
[37] Y. Liu, D. Cheng, Q. Wang, Q. Hou, L. Gu, H. Chen, T. Yang, Y. Wang, Optical distortion correction considering radial and tangential distortion rates defined by optical design, Results Opt. 3 (2021) 100072, https://doi.org/10.1016/j.rio.2021.100072.
[38] Z. Liu, X. Wang, C. Wang, W. Liu, X. Bai, Sparsetrack: multi-object tracking by performing scene decomposition based on pseudo-depth, arXiv preprint (2023), https://doi.org/10.48550/arXiv.2306.05238.
[39] W. Lv, N. Zhang, J. Zhang, D. Zeng, One-shot multiple object tracking with robust ID preservation, IEEE Trans. Circuits Syst. Vid. Technol. 34 (2024) 4473–4488, https://doi.org/10.1109/TCSVT.2023.3339609.
[40] V.R. Merenda, V.U.C. Bodempudi, M.D. Pairis-Garcia, G. Li, Development and validation of machine-learning models for monitoring individual behaviors in group-housed broiler chickens, Poult. Sci. 103 (2024) 104374, https://doi.org/10.1016/j.psj.2024.104374.
[41] A. Nasiri, Y. Zhao, H. Gan, Automated detection and counting of broiler behaviors using a video recognition system, Comput. Electron. Agric. 221 (2024) 108930, https://doi.org/10.1016/j.compag.2024.108930.
[42] S. Neethirajan, ChickTrack – A quantitative tracking tool for measuring chicken activity, Measurement 191 (2022) 110819, https://doi.org/10.1016/j.measurement.2022.110819.
[43] F.G. Proudfoot, H.W. Hulan, D.R. Ramey, The effect of four stocking densities on broiler carcass grade, the incidence of breast blisters, and other performance Traits1, Poult. Sci. 58 (1979) 791–793, https://doi.org/10.3382/ps.0580791.
[44] G.A. Quintana-Ospina, M.C. Alfaro-Wisaquillo, E.O. Oviedo-Rondon, J.R. Ruiz-Ramirez, L.C. Bernal-Arango, G.D. Martinez-Bernal, Data analytics of broiler growth dynamics and feed conversion ratio of broilers raised to 35 d under commercial tropical conditions, Animals 13 (2023), https://doi.org/10.3390/ani13152447.
[45] L. Riaboff, S. Poggi, A. Madouasse, S. Couvreur, S. Aubin, N. Bédère, E. Goumand, A. Chauvin, G. Plantier, Development of a methodological framework for a robust prediction of the main behaviours of dairy cows using a combination of machine learning algorithms on accelerometer data, Comput. Electron. Agric. 169 (2020) 105179, https://doi.org/10.1016/j.compag.2019.105179.
[46] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, IJCV 128 (2020) 336–359, https://doi.org/10.1007/s11263-019-01228-7.
[47] K. Simonyan, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv 1409 (2014) 1556.
[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[49] M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, Paper presented at the, in: Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2019.
[50] Ultralytics, Ultralytics YOLO Docs, Accessed on October 2024. Aviable at, https://docs.ultralytics.com/, 2024.
[51] Y. Wang, SMILEtrack: SiMIlarity LEarning for multiple object tracking, arXiv preprint (2022), https://doi.org/10.48550/arXiv.2211.08824.
[52] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 3645–3649.
[53] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, ByteTrack: multi-object tracking by associating every detection box, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 1–21.
[54] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, FairMOT: on the fairness of detection and re-identification in multiple object tracking, IJCV 129 (2021) 3069–3087, https://doi.org/10.1007/s11263-021-01513-4.