Automated Monitoring of Ear Biting in Pigs by Tracking Individuals and Events

Anicetus Odo
Queen's University Belfast
School of Biological Sciences
United Kingdom
a.odo@qub.ac.uk

Niall McLaughlin
Queen's University Belfast
School of EEECS
United Kingdom
n.mclaughlin@qub.ac.uk

Ilias Kyriazakis
Queen's University Belfast
School of Biological Sciences
United Kingdom
i.kyriazakis@qub.ac.uk

Abstract

We propose a system for automated monitoring of ear-biting in pigs. Ear-biting presents a welfare challenge to commercial pig farming, leading to injuries and infections that affect animal welfare. We use a computer vision system to detect and track all pigs and ear-biting events. Our goal is to provide early warning of ear-biting to allow quick intervention to improve the health and welfare of commercial farm animals. We compare several different object detection methods for the detection of individual pigs, including an oriented bounding box detector, which is better suited to the accurate detection of pigs from overhead cameras. We track all pigs and all ear-biting events using a specialised two-stage multi-object tracking system. The tracking system is adapted to match the characteristics of each entity being tracked. The tracking system allows the individual pigs involved in an ear-biting incident to be identified, allowing for targeted welfare interventions. We evaluate our complete system on real farm videos and demonstrate that our complete system improves compared to existing ear-biting detection methods.

1. Introduction

Pigs, which produce the most globally consumed meat, are usually managed under intensive conditions, which may lead to stress, aggression and behavioural abnormalities, such as vice. There is a global effort to enhance pig welfare through the development of improved housing, the adoption of enrichment and responsible farming practices, such as reduction in antimicrobial use. One way of achieving this is by monitoring pigs and acting early before problems are exacerbated. Monitoring pigs manually on large farms can be incredibly challenging due to the number of animals involved and the need for continuous attention. Pigs might hide or be difficult to spot in certain areas of the facility, making it challenging to identify sick or injured animals. Manual observations are subject to human error, and some health issues might not be immediately apparent to an observer. Many modern farms are turning to automation to aid such monitoring. This can include using cameras and sensors to track pig behaviour, health, and movement. Computer vision systems may be able to detect abnormal behaviours or signs of distress.

Contact behaviours that may lead to serious welfare challenges in pig management include tail and ear-biting. Recent approaches to detect and quantify these behaviours estimate interaction at the group level. Alameer et al. [1] detected tail biting by computing the interaction of group-housed pigs. Similarly, [17] introduced automated ear-biting detection at the group level. While automated systems that detect welfare challenges at the group level, such as ear or tail biting, are valuable tools for identifying overall issues in a pig population, raising an early warning and being able to link these challenges to individual pigs is crucial for effective intervention and management. Individual identification and tracking enable farmers to provide targeted mitigation and address specific welfare concerns.

Identification of individual pigs is critical for traceability and to enable the association of detected events with the performer. Several technologies are in use to help identify pigs with automated systems. For example, radio-frequency identification (RFID) tags can be attached to each pig and read by sensors around the facility, allowing events to be ascribed to individuals. The use of tags faces widespread resistance on the grounds of cost, handling at the various stages of production, and welfare. Alternative approaches may involve computer vision to recognise and track individual pigs based on unique features, markings, or patterns.

In this paper, we present a novel approach to detect ear-biting, a major vice with consequences in pig systems [24]. Our method does not require the use of tags, physical devices or markings. We use computer vision to associate ear-biting events with pigs in proximity. The main contributions of this paper include the following:

- We propose a method to generate a dataset of oriented bounding boxes that makes use of the Segment Anything Model (SAM) to reduce manual labelling effort. The oriented bounding box dataset is then used to train a fast, oriented pig detection model.

- We design an accurate multi-object tracking system for tracking pigs and ear-biting events. The system is evaluated with both axis-aligned and oriented bounding boxes.

- Our ear-biting detection system is evaluated on real farm data. Our complete system is capable of real-time operation. We achieve superior performance compared to existing methods in the literature. And we can reliably link ear-biting events with individual pigs to allow for targeted welfare interventions.

2. Related Work

Pigs are raised in intensive systems with limited space, which can lead to stress, aggression, and behavioural abnormalities. Tail biting is a serious welfare challenge in pig rearing [6,23]. Farmers resort to tail docking and teeth clipping to prevent tail-biting, and when tails are docked, aggression is redirected to other parts of the body, such as the ear. Ear-biting is a situation where one pig bites the ears of another pig, causing injuries and pain. An ethogram of ear-biting describing the behaviour of biter and bitten pigs was presented in [7]. Like tail biting, farmers typically learn about ear-biting long after it occurs, and this complicates the process of managing the associated diseases.

Video-based detection of welfare challenges in pigs has gained prominence and presents good application potential. Alameer et al., [1] studied tail biting and quantified the level of interactions in a pen using computer vision. They detected the heads and rears of pigs and defined tail biting as the contact between the head of one pig and the rear of another pig. Ear-biting is also a contact behaviour, but not all hear-to-head are ear-biting. An automated method for the detection of ear-biting using a sequence of snout-to-ear contacts was introduced in [17]. The approach quantifies ear-biting at the group level. They identified the need to identify consistent offenders (biter pigs), which would require continuous observations for longer periods of time. The identification of pigs may lead to management interventions, such as the removal of injured pigs for recovery.

While the detection of pigs and contact behaviours are critical for the continuous monitoring of pig welfare, accurate tracking of individual pigs over a long period of time will facilitate effective management of welfare concerns. Object tracking is a challenging subject in computer vision. Once the position of a pig is known in a frame, tracking helps to identify the same pig in consecutive video frames automatically. In pig management, a Multiple-Object-Tracking (MOT) identifies all the animals simultaneously. Several MOTs have been introduced, and the DeepSORT [27] is notable for its ability to learn features corresponding to the same identity in consecutive frames. The challenge of pig identification arises from a number of sources: (a) the direction of movement is completely random, (b) pigs usually stay close to each other for body warmth, (c) the animals look alike. These can result in loss of tracks and frequent identity switches.

In [18], the tracking problem was presented as a graph in which nodes are pigs and edges represent links to previous frames. Edges were classified to associate pigs across multiple frames. Re-identification of pigs after disappearance from the field of view is required for long-term tracking [26]. Schmidt et al. [20] use 3D images and identify each pig by observing the location of each pig's left and right ear, followed by cropping to provide ear tag images. Accurate segmentation of pigs in a pen depends on the level of occlusion, which is a factor of camera viewpoint and stocking density. Shuqin et al. [22] proposed instance segmentation for pig identification in a crowded environment. The technique fuses an MS R-CNN model with an adversarial network with the aim of effectively finding the boundary between pigs. The Segment Anything Model (SAM) [11] is a robust instance segmentation network that has achieved state-of-the-art performance in many computer vision applications. The method works well when prompted with texts, points, or boxes describing the regions of interest.

For all tracking-by-detection methods, the accuracy of the detector is important, as consecutive misses would affect the performance of the tracking system. Popular object detection models like the YOLOv5 [10] and YOLOv7 [25] use bounding boxes that are aligned to the axis of the image. Such axis-aligned bounding boxes create regions that encompass the target and a significant amount of background, especially as the target objects rotate. The additional background information created by such axis-aligned bounding boxes may be counterproductive for long-term tracking. Several applications involving observation from aerial cameras, such as in pig management, require bounding boxes that not only fit the targets but rotate with the target to reduce the chances of adding too much background information. An example application [12] has demonstrated the effectiveness of rotated bounding boxes for the overhead detection of animals in arbitrary poses that present a challenge to conventional object detection algorithms.

3. Methods

In this section, we will give details of our system for ear-biting detection in pigs. To enable deployment in realistic situations, we design our system so that all components can run in real-time. The first stage consists of a pig detector for axis-aligned or oriented bounding boxes. In Section 3.1, we show how to use the Segment Anything Model (SAM) [11] to create a dataset of oriented bounding boxes for detector training. In Section 3.3, the real-time multi-object tracking system is introduced. Finally, in Section 3.4, the ear-biting detection system, consisting of an initial ear-biting detector, a tracking system and an association method, is explained.

3.1. Pig Detection

Object detection systems are usually designed to produce bounding boxes aligned with the image axis. This means the bounding boxes can only have perfectly horizontal or vertical perimeter lines. However, when pigs are viewed from an overhead camera, they can take any orientation on the ground plane. This means standard axis-aligned bounding boxes frequently include a large amount of background pixels and do not accurately represent the orientation of the pig. Precise orientation information is useful for accurate tracking, especially when using tracking by detection, where bounding-box overlap is an important association feature.

We, therefore, explore the use of a detector capable of predicting oriented bounding boxes [29] and compare this with a standard axis-aligned object detection [10]. For both detectors, we use a standard YOLOv5L [10] backbone network.

3.2. Oriented Bounding Box Dataset Creation

To train a real-time detector to predict oriented bounding boxes, we require an appropriate training dataset of ground-truth oriented bounding boxes. Creating such a dataset would require substantial manual labelling effort. We, therefore, propose a novel automated method to produce an oriented bounding box dataset. We assume a set of standard axis-aligned bounding boxes has been given as ground truth. Such annotations can be produced relatively easily with standard tools compared to oriented bounding boxes.

For each video frame, we use the Segment Anything Model (SAM) [11] to generate segmentation masks for all pigs by providing the axis-aligned ground-truth bounding box coordinates as input to the SAM. The SAM model then generates a segmentation mask for each pig. However, the segmentation masks may have defects due to factors such as occlusion. Our pipeline for creating a dataset of oriented bounding boxes must, therefore, take steps to deal with these defects. An overview of the pipeline is shown in Fig. 1.

The steps are: First, axis-aligned bounding boxes and their corresponding images are provided as input to SAM, which produces segmentation masks. The segmentation masks are converted to grayscale and thresholded. Using the OpenCV Library [4], the contours of each masked region are extracted. Due to occlusions, the contour of one pig may be split into several parts, as shown in Fig. 1 (b). A morphological closing operator [21] is used to combine the gaps in the contour mask. Finally, an oriented bounding box is fitted to the major and minor axis of each contour for each pig. Each oriented bounding box is represented by the coordinates of its four corners $(x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4)$ in clockwise order.

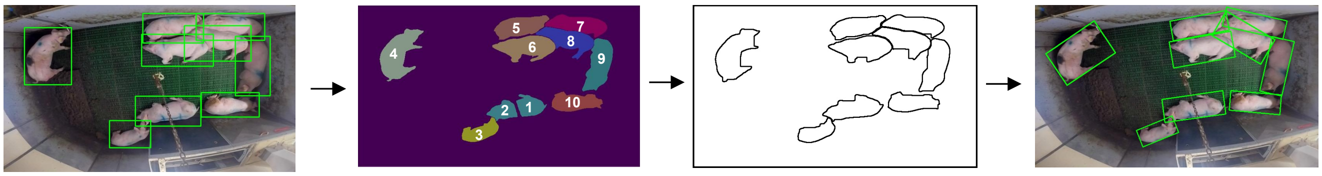
3.3. Multi-Object Tracker

The goal of the multi-object tracking system is to infer the position of individual pigs in all frames of the video. We use a tracker based on tracking-by-detection. This is a very efficient approach to tracking, which allows for faster than real-time operation. The tracker receives a set of detections for each frame of the video. The tracker must correctly link related detections over time to produce a set of tracks, recording the position of each pig at all times. Fig. 2 shows an overview of our multi-object tracking system.

Our tracker breaks up a given video into a sequence of short windows. The tracker is then composed of two stages. The first stage links detections within the current window into short, confident tracklets. The second stage then links the tracklets within the current window into longer, more confident tracks. Note that the second stage can be applied again to link tracklets between consecutive windows. Designing the tracker this way has the advantage that both past and future information can help improve tracking quality. This is especially important for tracklet linking in the second stage. Our approach contrasts with online trackers [5, 14], which must make correct linking decisions at every frame given limited information. The disadvantage of our approach is that we introduce a small amount of latency, of around 1 second, into the tracking process. For our application, this short latency period has no ill effects.

First Stage The first stage links all detections in the current window into short, confident tracklets. Given a window composed of frames, let $\mathcal{D}_f = (d_1, d_2, \ldots, d_D)$ be the set of detections at frame f. We have two possible bounding box detection formats, depending on whether the detector produces axis-aligned or oriented bounding boxes. Let us define an axis-aligned detection as $d_i = (x, y, w, h, c)$, which contains information on the position (x, y) and size (w, h) of each detection in pixel coordinates, and the detection confidence c. And let us define an oriented bounding box as $d_i = (x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, c)$, where the bounding box is defined by its four corner coordinates in pixel space in clockwise order. We do not use appearance information in the tracker, as all pigs tend to have a similar appearance. The tracker's first stage maintains a set of live tracklets $\mathcal{T}_l$, which are available for matching with new detections, and a set of finished tracklets $\mathcal{T}_f$, which cannot link with detections. Each tracklet is an ordered list of detections.

Video frames are processed sequentially. At each frame, every live tracklet's state, which includes its bounding box



(a) Ground-truth image with axis-aligned bounding boxes of all detected pig.

(b) Segmentation masks generated using SAM.

(c) Threshold output from SAM, close gaps in the binary image and create boundary of regions.

(d) Generate box-points for the minimum area oriented polygon of each contour.

Figure 1. **Oriented Bounding Box Dataset Creation** Axis-aligned ground truth bounding boxes are used as prompts for the Segment Anything Model [11]. Occlusions can introduce gaps in segmentation masks. A typical example is shown using labels 1 & 2, which belong to the same pig. The segmentation masks are cleaned up and used to create corresponding oriented bounding boxes.

position and size, is updated, assuming a linear motion model based on the average motion of the tracklet up to this point. At every frame, an association between live tracklets and detections in the frame is performed by calculating the pairwise affinity matrix $\mathcal{A}$ between every live tracklet and detection in the frame. The affinity cost $\mathcal{A}_{t,d} = 1 - IoU(t, d)$ between live tracklet t and detection d, is based on the intersection over Union (IoU) between the detection bounding box and tracklet bounding box [3]. Linking is permitted if the affinity cost is less than a threshold $\mathbf{T}_{iou1}$. The optimal association between live tracklets and detections is found by solving a linear assignment problem (LAP) [19]. LAP finds the minimum cost assignment between tracklets and detections based on the costs in the linking affinity matrix $\mathcal{A}$. Note that we do not use appearance information for association. We have found that pigs tend to have a similar appearance when viewed from our cameras. Therefore, simple appearance features, such as those based on colour histogram, do not adequately distinguish between individual pigs. We leave the exploration of more advanced appearance models, such as those based on re-identification features [26], for future work.

After association, we perform tracklet management. Any tracklets of length greater than a threshold $\mathbf{T}_{minlen1}$ that do not find a link at a given frame are immediately moved to the finished tracklet set. This helps to prevent false positive linking. These tracklets will be made available for linking by the second stage. Tracklets less $\mathbf{T}_{minlen1}$ frames long that do not find a link are deleted. Any detections not associated with tracklets at this frame start new tracklets available for linking at the next frame. If the end of the sequence is reached, all remaining live tracklets are moved to the finished tracklets set.

Second Stage Given the set of finished tracklets, $\mathcal{T}_f$, from the first stage, the second stage links these tracklets into longer confident tracks. For every pair of tracklets (t_i, t_j) in $\mathcal{T}_f$, we first check whether linking is permitted. We define the time in frames between the end of t_i and the start of t_j as δ frames. Linking between a pair of tracklets is permitted if δ is greater than zero and less than a threshold $\mathbf{T}_\delta$. If linking is permitted, we compute an affinity cost $\mathcal{A}_{t_i,t_j}$. The affinity cost is based on the IoU between the predicted state of each tracklet and the time difference. Assuming a linear motion model based on the previous state of each tracklet, we project the state of the older tracklet, t_i, forward in time by δ frames, and we project the state of the younger tracklet backwards in time by $-\delta$ frames. We then compute the average IoU between the predicted state of each tracklet and the corresponding state at the start or end of the opposite tracklet in the pair. If the IoU is greater than a threshold, $\mathbf{T}_{\text{IoU2}}$, linking is permitted. The final affinity cost $\mathcal{A}_{t_i,t_j} = (1 - \text{IoU}) + (1 - \delta/\mathbf{T}_\delta)$. The affinity cost is, therefore lower for links between trackers that are closer in time and similar motion, leading to higher IoU after projection by a linear motion model.

As with the first stage, the final association between tracklets is computed by solving a LAP based on $\mathcal{A}$. Note - this procedure finds pairwise links between tracklets. However, in reality, a long track may be split into many shorter tracklets. We, therefore, repeat the tracklet linking process until no further links are made. Each time, the tracklets tend to grow longer. As a final post-processing state, tracklets that are less than $\mathbf{T}_{\text{minlen2}}$ frames are discarded.

3.4. Ear-Biting Detection

The ear-biting detector consists of two parts. We first use an object detector to find low-level evidence of ear-biting, as has been done previously in the literature [17]. At each video frame, the detector produces a set of bounding boxes centred on potential ear-biting events. The detector is trained to find cases when the heads of two pigs are in a posture where ear-biting is likely to occur. However, when this detector is used independently, it is prone to false positives and missed detections. We, therefore, apply a modified version of the tracker to the raw stream of ear-biting-

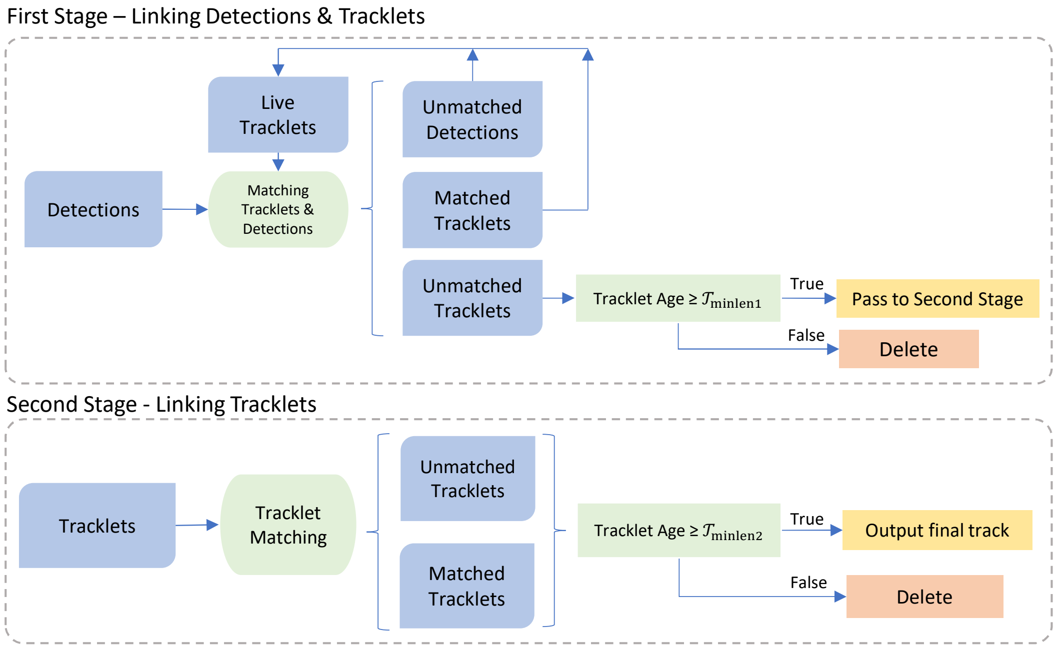


Figure 2. **Tracker Overview** (Top) First Stage - Links detections within a given window into short confident tracklets. (Bottom) Second Stage - Links short tracklets into longer, more confident tracks. The second stage can be applied again to link tracklets between windows.

event bounding boxes. By tracking all potential ear-biting events, we can compensate for both missed detections and false positives. The final output is a set of temporally coherent ear-biting events matched with the pigs involved.

Object Detector We used an existing method [17] to detect evidence of ear-biting. The method finds cases of close contact between the ear of one pig and the snout of another. However, the raw output from the detector is prone to generating false positive alerts. False positives occur because pigs are kept in close quarters and will frequently be in positions that trigger the ear-biting detector purely by chance.

Event Tracking The initial object detection stage produces a set of zero or more candidate ear-biting detection bounding boxes for every frame. These detections will include false positives and missed detections. We must, therefore, filter the raw detections so that only real events are included. To do this, we use the tracker. The tracker is already capable of associating bounding boxes over time and of coping with false positives and missed detections. The output will be a set of tracks, one for every real ear-biting event, recording the position of every ear-biting event at every frame.

We use a modified version of the tracker to track ear-biting events. This is because the ear-biting bounding boxes are quite different in size and motion from the pig bounding boxes. To improve tracking accuracy, we use C-BIoU [28]. This creates a buffer around every ear-biting bounding box, which increases the area where the association between ear-biting event detections in neighbouring frames can occur. This method makes matching ear-biting event detections easier despite their irregular motion and relatively small sizes.

Pig to Event Association Given the output from the multi-object pig-tracker (See Section 3.3) and ear-biting event-tracker, we can associate ear-biting events with specific pigs. This helps to identify pigs involved with events so that targeted welfare interventions can be made.

Given the set of pig tracks $P = (p_1, \ldots, p_N)$ and a set of ear-biting event tracks $B = (b_1, \ldots, b_K)$, we calculate the association between pigs and events. Given pig track p_n and event track b_k, we find the matching cost $1 - IoU(p_n, b_k)$, where the function $IoU(p_n, b_k)$ calculates the average Intersection over Union (IoU) between p_n and b_k in all frames where both tracks exist. A cost matrix $\hat{\mathcal{A}}_{NK}$ describes the cost of all possible associations. A threshold $\mathbf{T}_a$ is applied to remove weak matches.

The optimal association between pigs and events is found by solving a LAP. Note that every event is expected to have two associated pigs. Therefore, we first solve the LAP to find the pig most strongly associated with each ear-biting event. We then remove the column corresponding to this pig from the assignment matrix and solve the LAP again to identify the next pig associated with the event. Note that if an event is only associated with a single pig, it can be removed as a false positive.

4. Experiments

In this section, we evaluate the performance of our ear-biting detection system. We use several different datasets in our work. Our first dataset, used for training the pig detector and the initial ear-biting detector, has 6,466 (training images) and 803 (validation images). Each image has an average of ten pigs. Images were labelled with axis-aligned bounding boxes for individual pigs and ear-biting events.

Pig Detection Dataset We used 4,672 images from an existing dataset [16, 17]. In addition, 1,794 images from a different experiment [1] were added to the training set to represent the diversity of farm settings and features of pigs. We used 524 and 279 images from the respective datasets for validation. All the images were manually annotated with pig bounding boxes using the Visual Geometry Group (VGG) Image Annotator [8]. Table 1 shows the distribution of images and annotations.

Tracking and Ear Biting Dataset Our dataset, used for training and evaluation of the tracker and overall system evaluation, consisted of three videos recorded on real farms (Table 2). Each video was recorded from an overhead camera looking straight down so that all pigs could be seen at all times. The videos were collected from different rooms with varying levels of management challenges. A total of 1,045, 1,830 and 1,000 video frames were manually annotated with the location of each pig and their track numbers. Similarly, ear-biting event instances and tracks were created for system evaluation.

Detection Model Hyperparameters For pig detection, all input images were set to 640 x 640 pixels for training and testing. Both detection models were trained for 90 epochs using the Stochastic Gradient Descent (SGD) with a learning rate of 0.01, momentum of 0.937 and decay of 0.0005. Data augmentation hyper-parameters set for training include horizontal flipping with a probability of 0.5, mosaic with a probability of 1.0 and scale with a probability of 0.5. All other hyperparameters were set to default values for the YOLO network. Both networks were fine-tuned from the MS-COCO pre-trained weights, which was necessary due to the small size of our training set.

4.1. Pig Detection

In this section, we evaluate the performance of two pig detectors that differ mainly on the type of labels used (axis-aligned bounding boxes and oriented bounding boxes). We compare detection using standard object detection metrics

Data Set	Images	Pig boxes
Training	6,466	59,343
Validation	803	7,153

Table 1. Distribution of images and labels used for training and validation of pig detection.

Video	Images	Pig Boxes	Event Boxes
1	1,045	10,319	453
2	1,830	18,160	477
3	1,000	10,000	530

Table 2. Distribution of images and ground-truth bounding box labels for testing pig tracking and ear-biting events.

Video	Precision	Recall	AP@.5	AP@.5:.95
1	0.974	0.931	0.969	0.715
2	0.990	0.968	0.993	0.719
3	0.999	0.988	0.995	0.696

Table 3. Pig detection results using axis-aligned bounding boxes.

Video	Precision	Recall	AP@.5	AP@.5:.95
1	0.990	0.855	0.918	0.438
2	0.973	0.890	0.930	0.733
3	0.986	0.973	0.980	0.466

Table 4. Pig detection results using oriented bounding boxes.

for evaluation, including precision, recall and average precision (AP). Ground-truth boxes are axis-aligned. All regions with detection confidence ≥ 0.5 and Intersection-over-Union $(IoU) \geq 0.5$ were used for evaluation.

Both pig detectors are based on YOLOv5 [10]. This network has approximately 46.5 Million parameters. It takes around 10.1 ms to process each frame, meaning either detector is capable of running in real-time. The detectors achieved comparable AP on the validation set (0.987 and 0.964). Tables 3 & 4 show a summary of performance on the three test video sequences (Table 2). As shown, detections based on axis-aligned labels show slightly higher AP values. However, the objective of our approach is to minimize the amount of background information in detected pig regions, which could impact positively on tracking. Fig. 3 shows examples of pig detections from both systems. As shown, pig detection using axis-aligned bounding boxes (Left) capture significant amount of background information compared to oriented aligned boxes (Right).

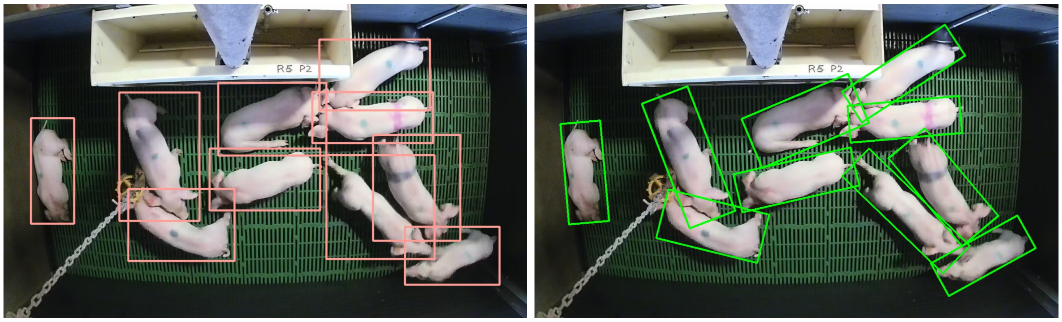


Figure 3. Detection of pigs comparing horizontally aligned bounding boxes and bounding boxes that rotate with targeted pigs (the left frame - showing regions based on the axis-aligned method and the right showing oriented bounding box based detections).

4.2. Multi-Object Tracking

In this experiment, we measure the performance of the multi-object tracker for the task of pig tracking. We compare the performance of our proposed tracker with Deep SORT, a widely used tracker from the literature [5]. We repeat this comparison for both axis-aligned and oriented bounding boxes. We measure tracker performance using the CLEAR-MOT metrics [15], which are the accepted standard in the academic literature for the evaluation of multiple object trackers.

Our tracker has a number of hyperparameters. We optimise the values of the hyperparameters and report results after cross-validation. To do this, we take the three labelled sequences and then treat two sequences as a training set and the third as a validation set. We use evolutionary search [9] (64 generations with a population size of 160) to find the tracker hyperparameters that give the optimal MOTA on the two training set sequences. We then measure performance on the held-out validation-set sequence. We repeat this process three times, using all three combinations of possible training and validation sets. This is a three-fold leave-one-out cross-validation. Finally, we report the average performance across the three validation set sequences.

Our tracker runs at 1883 fps with axis-aligned bounding boxes and 183 fps with oriented bounding boxes. Both are faster than real-time. The reduced speed of tracking with oriented bounding boxes is due to the more complex algorithm needed to compute IoU. Overall results in MOT-CLEAR format are shown in Table 5. We can see that for the DeepSORT tracker, the use of oriented bounding boxes leads to better tracking performance. This can be seen in the improved MOTA scores when oriented bounding boxes are used. For our proposed tracker, both kinds of bounding boxes give similar MOTA results, with axis-aligned having slightly better MOTA. However, we can see that the oriented bounding boxes have fewer ID switches and fewer false positives than the axis-aligned bounding boxes. And the IDF1 score, which focuses on association accuracy [13], is better when using oriented bounding boxes. Overall, our proposed tracker is better across all metrics compared to the DeepSORT tracker. We can see that when oriented bounding boxes are used, our tracker is generally better at correctly maintaining identity labels across time. The fact that our proposed tracker has a much lower number of ID Switches is important for the ear-biting detection problem. Ideally, we would like as few ID switches as possible so individual pigs can be accurately monitored over time to allow for targeted welfare interventions.

	DeepSORT [5]		Ours	
	Aligned	Oriented	Aligned	Oriented
↓ False Pos.	349	400	235	127
↓ False Neg.	4314	3587	720	810
↓ ID Switches	44	52	7	5
↑ Recall	88.80%	90.70%	94.51%	93.88%
↑ Precision	99.00%	98.90%	98.51%	98.95%
↑ IDF1	82.20%	81.70%	83.22%	89.35%
↑ MOTA	87.80%	89.50%	93.04%	92.81%

Table 5. Comparison between our proposed tracker and DeepSORT [5] used with either axis-aligned (Aligned) or oriented bounding boxes. Results are reported using the CLEAR-MOT tracker metrics [15]. For our system, we report the mean figures after three-fold leave-one-out cross-validation. For each statistic, we indicate whether higher or lower is desirable.

4.3. Ear Biting Detection

We compare the performance of our ear-biting detection system with an existing method in the literature [17]. As with the previous experiment on tracking (See Section 4.2), we use three-fold leave-one-out cross-validation. We then report the average results across all three validation sequences. The final ear-biting detection results are reported in Table 6.

We can see that our proposed ear-biting detection system

	Existing System [17]	Ours
↓ False Pos.	605	74
↓ False Neg.	524	219
↓ ID Switches	931	4
↑ Recall	64.10%	55.99%
↑ Precision	60.70%	83.80%

Table 6. Ear-biting detection performance. Comparison between our proposed system and an existing method from the literature [17]. For our system, we report the mean figures after three-fold leave-one-out cross-validation. For each statistic, we indicate whether higher or lower is desirable.

achieves a much lower number of false positives, false negatives and ID switches compared with the existing system. We can also see that its precision is higher. These statistics indicate that our system is better able to detect ear-biting events correctly, with a lower number of false alerts. A low number of false positives is important because any system that generates a large number of false positives alerts will be more likely to cause annoyance to end users. This can lead to alert fatigue, meaning the system may eventually be disconnected or ignored [2]. In a production system, reducing the number of false positives while maintaining high overall detection performance would be important and desirable.

4.4. Qualitative Results

We finally show the performance of our system at associating pigs with specific events. We show qualitative results from our proposed system in Fig. 4. We can see that two pigs are involved in an ear-biting incident. Their identities are maintained over a large number of frames and the ear-biting incident is recorded over a long period of time. We can also see how our system is capable of tracking multiple simultaneous ear-biting incidents. In a practical setting, individual statistics about involvement in ear-biting incidents would be maintained for each pig. This would allow for targeted interventions to improve animal welfare.

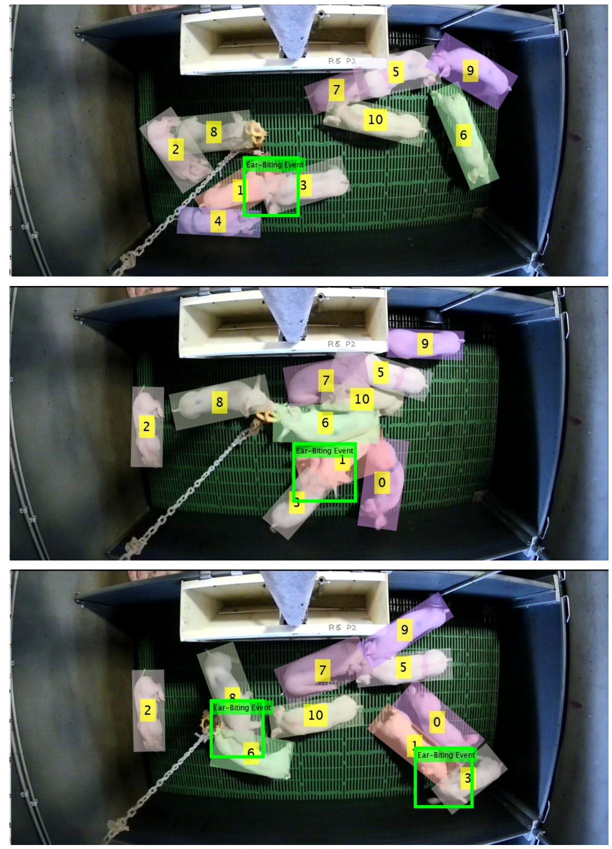


Figure 4. Qualitative results from our ear-biting detection system. We see two pigs, labelled 1 and 3, involved in an initial ear-biting incident. Both the pigs and the ear-biting event are correctly tracked over many frames. A second incident involving pigs 6 and 8 is also detected. (Top $t = 4s$, Middle $t = 24s$, Bottom $t = 33s$).

5. Conclusions

In this paper we have proposed a method to effectively monitor ear-biting incidents in pigs. Leveraging computer vision techniques, we developed a novel methodology for identifying pigs involved during ear-biting incidence. Our method used oriented bounding boxes for pig detection, which has significantly enhanced the precision of ear-biting detection for welfare monitoring. The process of data labelling is a crucial aspect of our method. We integrated the cutting-edge Segment Anything method. This innovative approach has substantially reduced the time and effort required for labeling datasets of farm animals. Specifically, we have introduced a novel pig tracking algorithm designed to facilitate event and pig tracking. This tracking algorithm, characterized by its robustness, has not only improved the accuracy of farm animal tracking from overhead cameras but has also effectively minimized the occurrence of false positive ear-biting events. One of the key achievements of our system is its ability to associate ear-biting events with specific interacting pigs during designated inspection windows. This capability enables timely intervention and a deeper understanding of the dynamics of such behaviors, ultimately contributing to improved pig welfare management.

Acknowledgement

This work was in part funded by the UK Biotechnology and Biological Sciences Research Council (BBSRC) Impact Accelerator Account of Queen's University Belfast.

References

[1] Ali Alameer, Stephanie Buijs, Niamh O'Connell, Luke Dalton, Mona Larsen, Lene Pedersen, and Ilias Kyriazakis. Au-

tomated detection and quantification of contact behaviour in pigs using deep learning. *Biosystems Engineering*, 224:118–130, 2022.

[2] Tao Ban, Ndichu Samuel, Takeshi Takahashi, and Daisuke Inoue. Combat Security Alert Fatigue with AI-Assisted Techniques. In *Cyber Security Experimentation and Test Workshop*, CSET '21, page 9–16, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[4] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[5] Mikel Broström. BoxMOT: A collection of SOTA real-time, multi-object trackers for object detectors, 2023.

[6] Jen-Yun Chou, Rick B. D'Eath, Dale A. Sandercock, Natalie Waran, Amy Haigh, and Keelin O'Driscoll. Use of different wood types as environmental enrichment to manage tail biting in docked pigs in a commercial fully-slatted system. *Livestock Science*, 213:19–27, 2018.

[7] Alessia Diana, Lenn Carpentier, Deborah Piette, Laura Ann Boyle, Daniel Berckmans, and Tomas Norton. An ethogram of biter and bitten pigs during an ear biting event: first step in the development of a precision livestock farming tool. *Applied Animal Behaviour Science*, 215:26–36, 2019.

[8] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *27th ACM Int. Conf. on Multimedia*. ACM, 2019.

[9] Ahmed Fawzy Gad. Pygad: An intuitive genetic algorithm python library. *arXiv preprint arXiv:2106.06158*, 2021.

[10] Glenn Jocher. Yolov5 by ultralytics, 2020.

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. *arXiv:2304.02643*, 2023.

[12] Dong Liu, Andrea Parmiggiani, Eric Psota, Robert Fitzgerald, and Tomas Norton. Where's your head at? detecting the orientation and position of pigs with rotated bounding boxes. *Computers and Electronics in Agriculture*, 212:108099, 2023.

[13] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129:548–578, 2021.

[14] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep OC-SORT: Multi-Pedestrian Tracking by Adaptive Re-Identification. *arXiv preprint arXiv:2302.11813*, 2023.

[15] Anton Milan, Konrad Schindler, and Stefan Roth. Challenges of ground truth evaluation of multi-target tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 735–742, 2013.

[16] Anicetus Odo, Ramon Muns, Laura Boyle, and Ilias Kyriazakis. Ear biting in pigs - a welfare challenge, 2022.

[17] Anicetus Odo, Ramon Muns, Laura Boyle, and Ilias Kyriazakis. Video analysis using deep learning for automated quantification of ear biting in pigs. *IEEE Access*, 11:59744–59757, 2023.

[18] Andrea Parmiggiani, Dong Liu, Eric Psota, Robert Fitzgerald, and Tomas Norton. Don't get lost in the crowd: Graph convolutional network for online animal tracking in dense groups. *Computers and Electronics in Agriculture*, 212:108038, 2023.

[19] Lyle Ramshaw and Robert E Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 20, 2012.

[20] Ty B Schmidt, Jessica M Lancaster, Eric Psota, Benny E Mote, Lindsey E Hulbert, Aaron Holliday, Ruth Woiwode, and Lance C Pérez. Evaluation of a novel computer vision-based livestock monitoring system to identify and track specific behaviors of individual nursery pigs within a group-housed environment. *Translational Animal Science*, 6(3):txac082, 06 2022.

[21] Pierre Soille et al. *Morphological image analysis: principles and applications*, volume 2. Springer, 1999.

[22] Shuqin Tu, Qiantao Zeng, Haofeng Liu, Yun Liang, Xiaolong Liu, Lei Huang, and Zhengxin Huang. MaskDis R-CNN: An instance segmentation algorithm with adversarial network for herd pigs. *IET Image Processing*, n/a(n/a), 2023.

[23] Anna Valros and Mari Heinonen. Save the pig tail. *Porcine Health Management*, 1(1):1–7, 2015.

[24] Nienke Van Staaveren, Julia Adriana Calderón Díaz, Edgar Garcia Manzanilla, Alison Hanlon, and Laura Ann Boyle. Prevalence of welfare outcomes in the weaner and finisher stages of the production cycle on 31 irish pig farms. *Irish veterinary journal*, 71(1):1–9, 2018.

[25] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[26] Meiqing Wang, Mona L.V. Larsen, Dong Liu, Jeanet F.M. Winters, Jean-Loup Rault, and Tomas Norton. Towards re-identification for long-term tracking of group housed pigs. *Biosystems Engineering*, 222:71–81, 2022.

[27] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018.

[28] Fan Yang, Shigeyuki Odashima, Shoichi Masui, and Shan Jiang. Hard to track objects with irregular motions and similar appearances? make it easier by buffering the matching space. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4799–4808, 2023.

[29] Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. *CoRR*, abs/2003.05597, 2020.