

Received July 15, 2019, accepted July 30, 2019, date of publication August 5, 2019, date of current version August 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933060

Automated Individual Pig Localisation, Tracking and Behaviour Metric Extraction Using Deep Learning

JAKE COWTON^{1,2}, ILIAS KYRIAZAKIS², AND JAUME BACARDIT¹

¹School of Computing, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

²School of Natural and Environmental Sciences, Agriculture, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Corresponding author: Jake Cowton (j.cowton2@newcastle.ac.uk)

This work was supported by the European Commission under the European Union Framework Programme for Research and Innovation Horizon 2020 under Grant 633531. The work of J. Bacardit was supported by the Engineering and Physical Science Research Council under Grant EP/N031962/1 and Grant EP/M020576/1.

ABSTRACT Individual pig tracking is key to stepping away from group-level treatment and towards individual pig care. By doing so we can monitor individual pig behaviour changes over time and use these as indicators of health and well-being, which, in turn, will assist in the early detection of disease allowing for earlier and more effective intervention. However, it is a much more computationally challenging than performing this task at group level; mistakes in identification and tracking accumulate and, over time, provide noise measures. We combine a deep CNN object localisation method, Faster Region-based convolutional neural network (R-CNN), with two potential real-time multi-object tracking methods in order to create a complete system that can autonomously localise and track individual pigs allowing for the extraction of metrics pertaining to individual pig behaviours from RGB cameras. We evaluate two different transfer learning strategies to adapt Faster R-CNN to our pig detection dataset that is more challenging than conventional tracking benchmark datasets. We are able to localise pigs in individual frames with 0.901 mean average precision (mAP), which then allows us to track individual pigs across video footage with 92% Multi-Object Tracking Accuracy (MOTA) and 73.4% Identity F1-Score (IDF1), and re-identify them after occlusions and dropped frames with 0.862 mAP (0.788 Rank 1 cumulative matching characteristic (CMC)). From these tracks we extract individual behavioural metrics for total distance travelled, time spent idle, and average speed with less than 0.015 mean squared error (MSE) for each. Changes in all these behavioural metrics have value in the detection of pig health and wellbeing.

INDEX TERMS Machine learning, re-identification, behaviour analysis, object detection, multi-object tracking.

I. INTRODUCTION

The ability to monitor the behaviour of animals, in particular, how said behaviour changes over time and under varying circumstances, provides us with knowledge that can assist in identifying problems before they become serious, or even life threatening, and enhances the success of intervention [1], [2]. Continuous physical monitoring of animals is impractical due to the effort required and the enormity of the scale of modern pig livestock units. As a consequence, farm staff usually resort to brief observations that are only

able to detect substantial changes or clinical signs. Thus they fail to detect subtle changes in behaviour that usually precede clinical signs of disease; this results in late intervention [1], [3].

We have previously developed a method that enables us to measure behavioural traits in groups of pigs [3]. Despite the usefulness of having group-level pig behaviour measures, there are several advantages in being able to automatically detect individual behaviours and identify pigs that may be at risk or are challenged [2] as this enables more personalised treatment plans. Targeted and selective animal treatment may result in furthering the trend towards reducing antimicrobial input in livestock systems [4]. However, individual-level

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

methods create data that is sparser and hence more sensitive to potential estimation errors [5]. Moreover, errors in pig identification and tracking can propagate to later frames, which is an issue that does not affect pig-level measurements. Here, we present an approach to detect and track individual pigs kept in groups, using inexpensive, colour (RGB) cameras in commercial farm conditions. The approach does not use individual pig identifiers, such as RFID, due to their impracticality and industry concerns over their use, mainly associated with their retrieval at the abattoir, potential residues and cost.

We make use of a Faster R-CNN [6] architecture adapted to our domain using transfer learning in order to detect the location of pigs in each frame of an RGB video. These detections are then used to generate identities that are then processed by multi-object tracking (MOT) algorithms. In this paper we present two approaches to tracking the identities of pigs between frames, one which uses trajectory-based prediction and association [7], and one which uses on both trajectory and similarity in visual appearance [8]. We then calculate behaviour measures from the tracks to illustrate how we can extract valuable and actionable knowledge from these algorithms in our concrete domain of application.

The main contribution of our method is that it provides a full workflow to localise, track, and extract behavioural metrics of individual pigs using only an RGB camera, in real-time, without the need for additional hardware (such as ID tags) or visual aids (such as IDs marked directly on pigs), whilst also still being feasible to install in a commercial farm. We also demonstrate that, despite the very similar visual appearance of the pigs in our dataset, deep learning methodologies can be used to generate feature vectors capable of discriminating between identities of pigs, which is key to understanding the potential applications of deep learning in precision livestock farming.

Section II-A describes the datasets that were used for training and evaluation in our models. Section II-B describes the detection method used followed by Section II-C which outlines the two methods used for multi-object tracking, and how the deep association metric model was trained. Section II-D describes how we use these tracks to extract behaviour metrics for individual pigs and is followed by Section II-E which describes how each of the components of our method are evaluated. Section III and Section IV outline and discuss the results of each of the components respectively. We finalise with the conclusion in Section V.

II. MATERIALS & METHODS

In this section we describe the data used to train and evaluate our methods for detection, tracking and extracting behavioural metrics, the model used to detect the location of pigs in an image, the tracking methods that were used to assign the detections to identities, and how we analysed the tracks to extract behavioural metrics. We also describe the means by which we evaluate these methods.

A. DATASET DESCRIPTIONS

Three standard datasets were used in order to implement all of the components of our work: ImageNet [9], an image classification dataset commonly used to pre-train a Convolutional Neural Network (CNN) that processes images; Pascal Visual Object Classes Challenge 2007 [10], commonly used as a benchmark dataset for object detection and classification; and Motion Analysis and Re-identification Set (MARS) [11], a video-based person re-identification (Re-ID) dataset. Additionally, we used three of our own, manually-annotated, pig datasets: one for detection (Section II-A.1), one for tracking (Section II-A.2), and one for re-identification (Section II-A.3).

We selected distance travelled, time spent idle, and average speed for behavioural metrics to extract from individual pig tracks as they best inform us about a pigs activity levels, an valuable metric for understand pig health and wellbeing.

The images used in these pig datasets were collected at Newcastle University's Cockle Park farm. Because the images were collected from pigs husbanded under farm conditions, there was no need for ethical approval. The building used houses 4 pens, each containing 20 pigs of the same age and of balanced weight. The pigs were recorded using the RGB sensor within the Microsoft Kinect v2 (resolution: 1920×1080 , field of view: 84.1×53.8 , focal length: 3.29, shutter speed: 14ms, max frame rate: 30 FPS) mounted on the ceiling of the room, pointing downwards. Although we used this specific sensor, for the purpose of the methods presented in this work, any RGB camera with similar parameters would be suitable as we do not use any of the specialised features of the Kinect. The camera was tilted at an angle (rather than perpendicular to the floor) so that the camera covered half of one of the pens. There was some overlap in the images with the adjacent pen (Figure 1), separated by a red wall, however we removed any detections in this area. Due to technical constraints of the data capturing infrastructure, we were not able to record long, continuous segments of video so were limited to theoretical maximum of 10 minutes, which, although short, is longer than a human observer would realistically spend watching the animals.

1) PIG DETECTION DATASET

Our manually-annotated (Figure 1) pig dataset consisted of 1,646 images (50% used for training and 50% used for testing) consisting of 9,014 annotations. All data used in this dataset came from the same camera and the same batch of pigs, though was collected across different days and times. The days used for creating the test set are not included in the training set in order to create a level of separation. This dataset contained roughly 6 times fewer images and nearly 3 times fewer annotations than Pascal Visual Object Classes Challenge 2007 (VOC), hence indicating that our dataset has a higher annotation density (number of objects per image). The pig detection dataset suffered from a number of complexities that are not found in the VOC dataset. The challenges largely stem from the fact the data was collected on a commercial



FIGURE 1. A example image from the pig dataset where pigs are densely packed into one area with corresponding ground-truth annotations.

farm environment. For example, the reason there were more annotations per image in this dataset was because the objects were often very densely packed into a small space within the image 1. This made separation of the pigs a very difficult task, even when carrying out the manual annotations.

Secondly, the quality of the images in the pig dataset was substantially lower than that of the images in VOC. This was because the images were captured on a live farm, which is extremely dusty. Additionally, images can sometimes be overexposed due to natural light entering through windows. Having this natural light is a requirement in order to comply with UK legislation. When contrasted with the very high-quality level of photographs used in VOC there was a very significant difference. Because the features of an object were much more difficult to identify due to the lower image quality, models trained on this data are at a disadvantage as these are typically what are used in order to create a generalisable model.

These factors combined show that the pig dataset was a substantially more difficult dataset than VOC, which is something that previous research has had to overcome by employing post-processing methods [12]. In order to quantify how our implementation performs in these conditions, in addition to testing on the whole test set, we also assessed the detection performance independently for images containing: many pigs, densely packed pigs, overexposed images, and low-light images (Figure 2).

Images were classed as containing many pigs if more than 10 pigs were in the image, as this meant more than half the pigs were in less than half the pen space (4% of the test set). As for the densely packed pigs segment, images were placed here when more than 4 bounding boxes were overlapping (43% of test set). Images were determined to be overexposed by manual annotation (11% of test set). Finally, the low-light segment was made up of images where the average brightness of a pig was lower than 100 (4% of test set). Pig brightness was calculated by converting a bounding box containing a pig to greyscale and taking the average pixel intensity.

As can be seen in Figure 1, the camera's field of view overlapped with the adjacent pig pen. We therefore ignored any

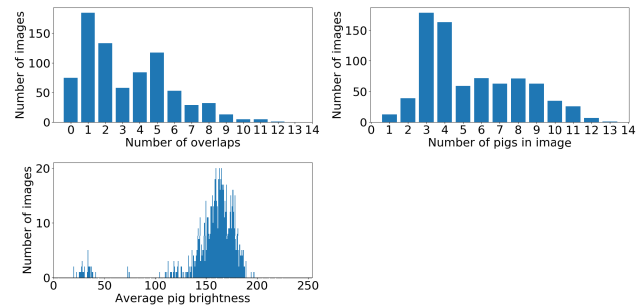


FIGURE 2. Distributions of the number of overlapping bounding boxes per image (top left), the number of pigs per image (top right) and the average brightness of a pig per image (bottom left) within the test set.

detections from this area of the camera. This was achieved by setting a threshold on the Y-axis, along the red wall, and ignoring any detections to the left of it.

2) PIG TRACKING DATASET

Images from the detection dataset were used to create another dataset specifically for tracking. This consisted of a single 7.8 minute video recorded at an average of 4 frames per second (FPS) (~1,874 frames) from a single camera covering half a pig pen (the maximum was 10 FPS). Each frame was annotated, in the same way as the detection dataset, however IDs were given to each pig that persisted from frame to frame. Due to the nature of the recording environment, in particular the hardware used, the recording varied in its FPS and regularly dropped frames throughout the recording. This resulted in some drastic changes, frame-to-frame, in some situations (e.g. a pig appear in the middle of the pen, or extremely quickly moving pigs).

Moreover, as pigs leave and later re-enter the scene, we cannot continuously track them for the whole length of the video. This has resulted in a set of 25 manually curated, unique tracking IDs. Some tracks last for most of the video while some others are fairly short. The duration of each track is represented in Figure 3.

3) PIG RE-IDENTIFICATION DATASET

In addition to this single-camera tracking dataset, we gathered a separate, dual-camera pig Re-ID dataset (Figure 5), structured similarly to MARS [11], a dataset for person Re-ID. This pig Re-ID dataset consisted of 25 pig identities, where each identity had an average of 280 images, totalling 5,653 images (60% for training, 40% for testing). All annotations were resized to be 128×256 for processing by a CNN (outlined in Section II-C.2). This is a difficult Re-ID dataset as, when compared to that of a person Re-ID dataset, such as MARS, where clothes strongly distinguish two people apart, pigs look very similar to one another (Figure 5).

B. PIG DETECTION METHOD

The Faster Regions with CNN features (Faster R-CNN) model used to detect pigs location consisted of 3 main

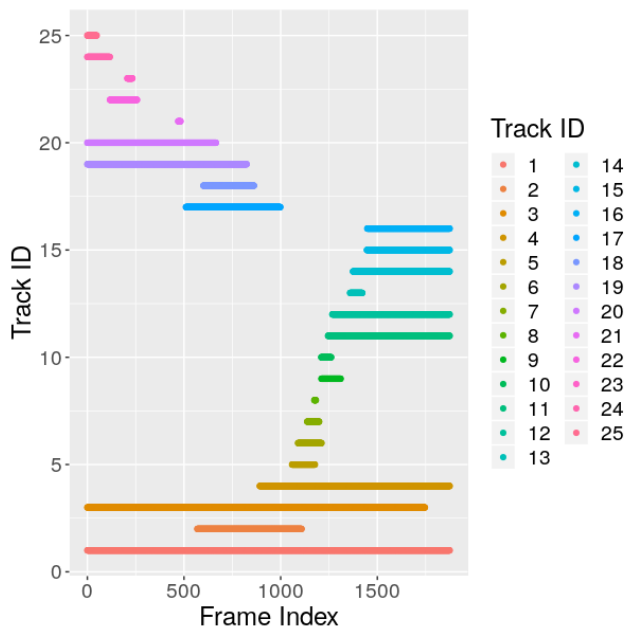


FIGURE 3. Representation of the manually annotated pig tracks. The Y-axis shows the ground truth pig ID, the X-axis shows the frames during which the pig was visible. Once a pig left the camera, it was not re-identified, and was therefore given a new ID.



FIGURE 4. Top: A sample of two identities of the MARS dataset for person re-identification. Bottom: A sample of two identities of the pig Re-ID dataset.

components: the feature extractor, the Region Proposal Network (RPN) and the fully-connected (FC) layers. The first was responsible for creating a fixed-length vector from an input image, the second proposed regions that are likely to contain a pig, and the final component classified these regions into a class (e.g. background or pig for the pig dataset). ResNet-101 [13] was used for the feature extraction in all implementations of Faster R-CNN.

For the training of our models we used the parameters that perform best on the VOC dataset (Table 1) using stochastic gradient descent. An Nvidia Tesla P100 graphics card was used for both training and inference.

We made use of all three datasets in order to build this model. The feature extraction layers of the model were pre-trained on ImageNet. This is common when training any CNN on images as the dataset is extremely large (~14 million images) and therefore takes a substantial

TABLE 1. The parameters used for the Faster Regions with CNN features that perform best on the Pascal Visual Object Classes Challenge 2007 dataset.

Learning rate (LR)	0.001
LR Decay step	5
LR Decay multiplier	0.1
Batch Size	8

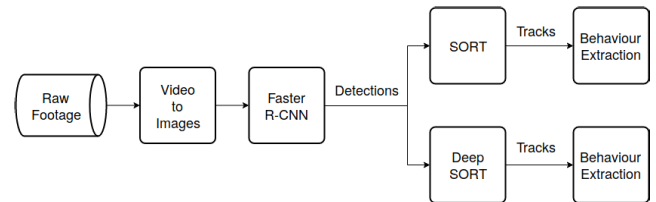


FIGURE 5. A breakdown of the full workflow of our implementation from the video footage of a pig pen, to the behavioural metrics we extract from the tracking methods.

amount of time to train from randomly initialised weights (a week in some cases, dependent upon hardware). The RPN and FC layers of the Faster R-CNN were trained using two datasets: firstly VOC, followed by the pig detection dataset. It was expected that the densely-packed nature of the pigs at certain time points in the dataset would be particularly a problem for the Faster R-CNN as it uses non-maximum suppression (NMS) to filter overlapping bounding boxes which is typically enacted when Intersection over Union (IoU) > 0.7 [6] (described in Section II-E.1).

Before we could train the model on the pig dataset after training on VOC, it was necessary to modify the model architecture to account for the change in number of potential classes. This change in model architecture is referred to as transfer learning, where a model is trained to solve one task, in order to help it solve a different, but related, problem. There are multiple ways this can be implemented, two common approaches would be to train the RPN and FC layers on VOC, then modify the FC layers to account for the different classes in the pig dataset (20 in VOC, 1 in the pig dataset). Alternatively, an additional layer can be added to the final FC layer after training on VOC which has only one class outcome. This results in two output nodes (1 for pig and 1 for background), which creates in an additional 78 trainable parameters in our fully connected layers. Despite this slight increase in number of parameters, we used the transfer learning strategy that adds an extra layer as it performed better than the other evaluated strategies (Section III-A).

C. PIG TRACKING METHODS

Once the Faster R-CNN detected the pigs location in each frame, it was necessary to track the identity of each pig between frames. To achieve this, we evaluated two alternative strategies for this task.

1) DISTANCE-BASED TRACKING

We employed Simple Online and Real-time Tracking (SORT) [7], which combines the Kalman filter [14] with the

TABLE 2. An overview of the CNN architecture used to learn the association metric for pig re-identification. This is trained using MARS followed by fine-tuning on our own pig Re-ID dataset.

Layer No.	Name	Size/Stride	Output
1	Conv	$3 \times 3/1$	$32 \times 128 \times 64$
2	Conv	$3 \times 3/1$	$32 \times 128 \times 64$
3	Max Pool	$3 \times 3/2$	$32 \times 64 \times 32$
4	Residual	$3 \times 3/1$	$32 \times 64 \times 32$
5	Residual	$3 \times 3/1$	$32 \times 64 \times 32$
6	Residual	$3 \times 3/2$	$64 \times 32 \times 16$
7	Residual	$3 \times 3/1$	$64 \times 32 \times 16$
8	Residual	$3 \times 3/2$	$128 \times 16 \times 8$
9	Residual	$3 \times 3/1$	$128 \times 16 \times 8$
10	Dense	-	128
11	l_2 Norm	-	128

Hungarian algorithm [15] to create an improved multi-object tracking algorithm. This method is highly dependent upon accurate object detection, as only the location of objects is used, and is an entirely unsupervised method. As no training data is required, it can be applied directly to the detections found for the tracking test data set outlined in Section II-A.2.

2) DISTANCE & VISUAL-BASED TRACKING

We also made use of Deep Simple Online and Real-time Tracking (Deep SORT) [8], which, alongside the Kalman filter & Hungarian algorithm used in SORT, uses a learned association metric to determine if two images of a pig contain the same pig or not. In this method, when identities of objects are being matched between frames, both the trajectory prediction and the association metric are used to determine if two objects in different frames are the same. Because this method is able to identify if a “newly” identified pig actually belongs to an identity that has already been established, it is much more capable of handling long-term occlusions and corruptions in the dataset (e.g. dropped frames), which is not uncommon in the datasets we used. This method is configured to only use the previous 480 images (2 minutes at 4 FPS) so that it can consistently track a pig as it grows. If all previous known images of the pig were stored it could degrade performance as pigs change in appearance over time.

This association metric is learned using a deep learning CNN model (described in Table 2) which uses a re-parametrisation of the softmax classifier that includes a measure of cosine similarity in the representation space, which is a 1×128 vector, initially developed for person Re-ID [16]. Where an input is a cropped image of a detected object, the network is trained to minimise the cross-entropy loss of the class predictions generated by the model and the true label distribution. Once the generation of the feature vector has been learned, the classifier is discarded and the feature vectors produced can then be compared with the feature vectors that are stored for each existing identity using so that it can be assigned to the identity with the closest match that also appears within range of the predicted location produced

by the Kalman filter. Evaluation for this method is conducted using a single query image which is then matched with an image from a gallery of images. Where there are multiple cameras in the dataset, the gallery images are taken from different cameras.

The original implementation [16] of the CNN that generates the association metric was created for the purpose of person Re-ID. This research area focuses on being able to identify if two images or videos of a person are the same person. The key applications of this research focuses on facial recognition and datasets where the objective is to track individual people within a crowd, allowing for individuals to become temporarily occluded and then recovering their original tracking identity. The network was pre-trained on MARS followed by training using the pig Re-ID dataset (Section II-A.2) to optimise the method to be able to identify identical pigs in a commercial setting. The CNN was trained using the Adam optimiser [17] with a low learning rate of 0.00001 and a batch size of 128.

D. BEHAVIOURAL METRICS EXTRACTION

Once tracks were established for individual pigs, we were able to derive behavioural metrics pertaining to each individual track. We measured average speed, total distance covered and time spent idle as these values quantify how active pigs are. Change in activity has been linked to several pig health and welfare challenges, with the general trend being that such challenges tend to decrease the levels of activity in individuals [1], [3]. We track activity levels through measuring idle time, which is the reciprocal of active time.

In order to calculate distance travelled and idle time we used the Euclidean distance between the centre-point of each detection box from frame to frame. We did not convert these measurements into real-world distances as it is simpler to calculate and because we have variable FPS, which we did not store when recording the footage. This, for example, makes it not possible to accurately estimate speed. This causes some discrepancies, albeit small, as the camera we used was not perfectly perpendicular to the floor; it was set at an angle in order to have full coverage of the pen. The amount of time a pig spent idle was defined by the amount of time where the pig moves no more than 4 pixels between frames.

E. EVALUATION

1) DETECTION EVALUATION

Intersection over Union (Equation 1) was used to assess how accurate a predicted bounding box was in comparison with the ground truth (localisation performance), which was calculated using Equation 1. The higher the IoU, the more accurate the bounding box is (Figure 6). Rather than using the threshold of 0.5 to determine whether the IoU of a predicted bounding box is accurate, which is the standard proscribed in the Pascal Visual Object Classes Challenge 2007 challenge, we required an $\text{IoU} \geq 0.6$.

$$\text{IoU} = \frac{\text{Area of overlap between bounding boxes}}{\text{Area of union between bounding boxes}} \quad (1)$$



FIGURE 6. Examples of how Intersection over Union is calculated. Left: Poor performance IoU = 0.4034. Middle: Good performance, IoU = 0.7330. Right: Excellent performance, IoU = 0.9264.

In order to summarise the performance of the detector, we used mean average precision (mAP) a standard metric that is used to evaluate the performance of object detection methods [6], [18]–[20].

2) DEEP ASSOCIATION METRIC EVALUATION

In order to evaluate the performance of the association metric learned by the CNN outlined in Section II-C.2, we made use of the overall mAP of the classifier and the Cumulative Matching Characteristic (CMC). It is common to evaluate CMC ranks 1 through to 20 [21], however, as we only have 25 identities in our pig Re-ID dataset, it was not possible to do this. As with many 1 : m identification systems, for our application, the CMC at rank 1 is the most crucial [22], [23]. We therefore focus predominantly on this metric, though we additionally report CMC ranks 3 and 5, in order to better show how our model is generally performing.

3) TRACKING EVALUATION

We used the widely accepted metrics outlined in the 2016 MOT Challenge [24] in order to evaluate the performance of our tracker, implemented using the py-motmetrics library [25]. The tracking performance measure we used was multi-object tracker accuracy (MOTA) (Equation 2), the most commonly used metric to benchmark MOT solutions, as it accounts for the three types of error that occur: false negative (FN), false positive (FP) and identity switch (IDSW). False negatives are defined as an object that is not tracked, false positives are defined as tracked objects which should not be and identity switches are when two objects that should be tracked swap identities. Fragmentations are defined as the number of times an identity switches from “tracked” to “not tracked”.

$$\text{MOTA} = 1 - \frac{\sum_t \text{FN}_t + \text{FP}_t + \text{IDSW}_t}{\sum_t \text{GT}_t} \quad (2)$$

However, as MOTA has shortcomings in terms of how it accounts for identity switches [26], we also report tracking metrics using IDF1, which is also included as a metric in the MOT Challenge, as this provides a much more global way to assess the performance of the tracking system in terms of its ability to track identities.

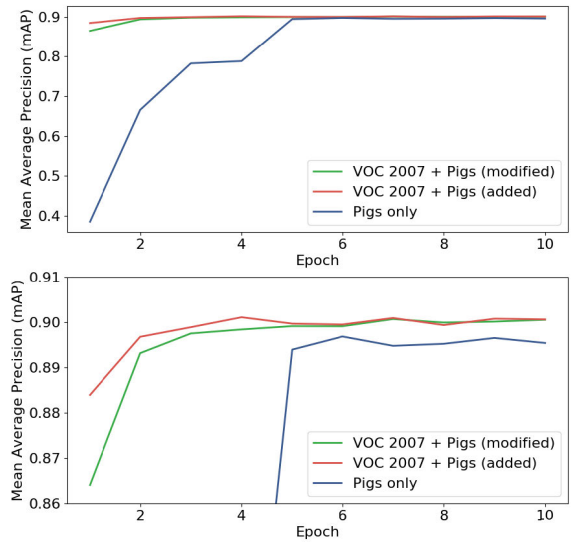


FIGURE 7. Top: Performance of Faster R-CNN models trained on a dataset of pigs in a live farm using 2 methods of transfer learning from a model pre-trained on Pascal Visual Object Classes Challenge 2007: adding an additional fully-connected layer and modifying the final fully-connected layers, along with a model trained only on the pig data. Bottom: The same data zoomed in.

4) BEHAVIOUR METRICS EXTRACTION EVALUATION

All detected tracks were grouped respective to the ground truth pig ID to which they belonged. For total distance travelled and time spent idle, the frame-to-frame estimations belonging to each pig were summed and speed was averaged. All behavioural metric estimations were evaluated by normalising the values to between 0 and 1, followed by calculating mean squared error (MSE) (Equation 3), where i is the pig ID, Y is the ground truth behaviour and \hat{Y} is the predicted behaviour.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (3)$$

Additionally, we calculated the absolute error for each of the behaviour metrics for each pig identity and carried out a paired Wilcoxon test [27] on them to determine if the absolute errors for each method were significantly different.

III. RESULTS

A. DETECTION RESULTS

Results from the Faster R-CNN (Figure 7) show that the performance difference between adding an FC layer or modifying the final FC layers was negligible for our application. This was also the case for detection (inference) speed, as both models had an average inference time of 80ms per frame. Both models that were pre-trained on VOC outperformed the model which was only trained on the pig data in terms of performance and speed of learning, hence showing the effectiveness of the transfer learning strategy.

The model that was only pre-trained on ImageNet (but not on VOC) achieved good results (mAP = 0.894) but



FIGURE 8. Four sample images from our pig detection test set processed by the Faster R-CNN with the feature extraction layers pre-trained on ImageNet, the rest pre-trained on Pascal Visual Object Classes Challenge 2007 and an additional fully-connected layer for the pig dataset. Detections to the left of the red wall are ignored. The top left image is from the low-light test segment. The top right image is from the densely packed test-segment. The bottom left image is from the overexposed test segment. The bottom right image is from the many pigs test segment.

TABLE 3. The parameters used for the Faster Regions with CNN features that perform best on the Pascal Visual Object Classes Challenge 2007 dataset.

Test Segment	mAP
Many pigs	0.905
Densely packed	0.906
Overexposed	0.906
Low-light	0.850

never reached the same level of performance as the other two models (mAP ~ 0.901). This is as expected as the other models are not only pre-trained on more data, but also on a related task, which is highly beneficial in transfer learning, but nonetheless the inclusion of this option in our comparison provides us with a useful performance baseline.

Figure 8 shows examples of detections made from each of the test segments defined in Section II-A.1. The top left image shows the Faster R-CNN correctly detecting 2 pigs from the low-light test segment. The top right image shows that the model was capable of detecting pigs from images in the densely packed pigs test segment. The bottom left image exemplifies issues relating to overexposure caused by strong sunlight which distorts the edges of the pigs, making them more difficult to detect. The model does appear to have suffered from the camera being at an angle rather than pointing directly down. This causes some pigs at the top of the images to be hidden behind other pigs in such a way that distorts their shape and makes them undetectable. This is particularly noticeable in the bottom right image, which is from the test segment for images containing many pigs.

In order to proceed with adding MOT tracking to our method, we selected the model which added an additional, final layer to the FC layers. We used this model to evaluate the test segments individually (Table 3).

Test segments with many, densely packed, and overexposed pigs performed in line with the rest of the dataset.

However, images that suffer from low-light relatively under-perform. This is mainly due to the fact that Faster R-CNN is an image based detection method, which requires light in order to detect objects. Nonetheless, the implementation does perform moderately well in low-light conditions.

B. ASSOCIATION METRIC LEARNING

As described in Section II-C.2, the CNN used to learn the association metric was trained using MARS followed by our own pig Re-ID dataset (Section II-A.3). Compared to the MARS dataset (1.1 million images), our pig Re-ID dataset is very small (5,653 images). Despite this, the fine-tuning increased the rank 1, 3, and 5 CMC by 17%, 15%, and 15% respectively, achieving a rank 1 CMC of 0.788. The additional training also increase the overall mAP from 0.760 to 0.862. This increase in performance is to be expected, as we are fine-tuning, but it is valuable to understand by how much the additional training has improved performance.

C. TRACKING RESULTS

As discussed, the SORT tracker is heavily dependent upon accurate detections. Therefore, as the detector was capable of achieving a high level of mAP (0.901), it was expected that the tracker would perform similarly well.

The direct output of the SORT tracker is a series of IDs, which then are mapped to our manually-annotated tracks (Section II-A.2). From this mapping process we can identify when the tracker detects an object that does not exist (FP), when the tracker is not able to detect an existing object (FN), or when the identifiers of two tracks are switched (IDSW) [24], [25], [28]. The result of this implementation was a large number of “tracklets” (partial tracks), subsets of which belong to individual pig identities.

SORT achieved a score of 95.1% MOTA. In total there were 153 FPs, 331 FNs, 50 IDSWs and 56 fragmentations. The average number of consecutive frames which perfectly tracked all pigs was 21.746 frames (5.437 seconds), with a maximum of 208 frames (52.000 seconds). The average length of time an individual pig could be perfectly tracked for was 129.358 frames (32.339 seconds), where the maximum was 981 frames (4 minutes). To further characterise the results of our method, Table 4 reports, for each of the 25 unique pig IDs, the percentage of frames for which such ID was correctly tracked. Five out of the 25 IDs had a perfect tracking score of 1.00, and 21 out of 25 had at least 0.9 successful tracking proportion. One ID had a very poor score of less than 0.5.

From the metrics derived, we can see that the general implementation works well (95% MOTA), but the occasional dropping of frames caused by the implemented recording system seriously impacts the continuity of IDs given to pigs between frames which is better represented by the 70.3% IDF1 score.

Unlike SORT, Deep SORT is less reliant upon accurate detections, though it does still require them to be of good quality as it still makes partial use of the Kalman filter to make assignment decisions between the existing tracklets

TABLE 4. Results of the SORT & Deep SORT tracking algorithm used to track individual pigs. ID is the ground truth ID for a pig, F is the ground truth for how many frames the pig was visible, T are the number of tracklets the method created for each individual pig, C is the percent of the ground truth tracks that were tracked by the method, S is the number of identity switches that occurred, FN is the number of false negatives (pig not detected). The arrows indicate whether lower or higher is better. There were also 153 and 105 total False Positives (a pig was detected that did not exist) for SORT and Deep SORT respectively.

ID	F	SORT				Deep SORT			
		T ↓	C ↑	S ↓	FN ↓	T ↓	C ↑	S ↓	FN ↓
1	1875	11	0.97	10	48	6	0.95	4	97
2	540	2	0.99	0	5	3	0.96	1	20
3	1747	6	0.98	4	33	6	0.96	4	73
4	983	2	1.00	0	2	2	0.99	0	8
5	119	2	0.99	0	1	4	0.92	2	8
6	122	3	0.95	1	5	3	0.80	1	23
7	63	1	1.00	0	0	2	0.92	0	5
8	9	3	0.33	1	5	2	0.33	0	6
9	100	2	0.97	0	3	3	0.91	1	8
10	52	1	1.00	0	0	1	1.00	0	0
11	630	7	0.97	5	16	5	0.92	3	48
12	608	18	0.79	16	110	7	0.81	5	111
13	64	4	0.83	2	9	4	0.61	2	23
14	501	2	0.99	0	3	2	0.98	0	9
15	429	3	0.94	1	25	4	0.87	2	53
16	427	6	0.90	4	38	5	0.69	3	130
17	489	1	1.00	0	0	1	1.00	0	0
18	263	3	0.96	1	9	3	0.92	1	20
19	825	3	1.00	1	3	3	0.97	1	22
20	666	5	0.98	3	9	4	0.96	2	25
21	13	2	0.77	0	3	2	0.08	0	12
22	141	3	0.98	1	2	3	0.94	1	7
23	27	2	0.93	0	2	2	0.70	0	8
24	117	1	1.00	0	0	2	0.92	0	9
25	49	1	1.00	0	0	2	0.82	0	9
Avg.		3.76	0.90	2	13.2	3.24	0.84	1.32	29.3

and detections from the following frame. Deep SORT achieved a score of 92.1% MOTA. In total there were 105 FPs, 734 FNs, 33 IDSWs, and 40 fragmentations. The average number of consecutive frames which perfectly tracked all pigs was 24.163 frames (6.041 seconds), with a maximum of 208 frames (52.000 seconds). The average length of time an individual pig could be perfectly tracked for was 197.882 frames (49.471 seconds), where the maximum was 981 frames (4 minutes). Two out of the 25 IDs had a perfect tracking score of 1.00, and 16 out of 25 had at least 0.9 successful tracking proportion. Two IDs had a very poor score of less than 0.5.

Although there was almost double the number of FNs, there was a 32% decrease in the average number of IDSWs and a 31% decrease in the number of false positives raised by the system. Despite the increase in FNs being substantial, and therefore also decreasing the proportion of track coverage, it was considered a fair trade-off, as a FN is much easier for the system to recover from than an IDSW. This is because an IDSW tends to be permanent, where an FN may only be for a few frames, after which, the identity can be recovered. Because of this ability to recover from FNs and the decrease in IDSWs the introduction of the visual Re-ID component within Deep SORT results in an increase in IDF1 to 73.4%

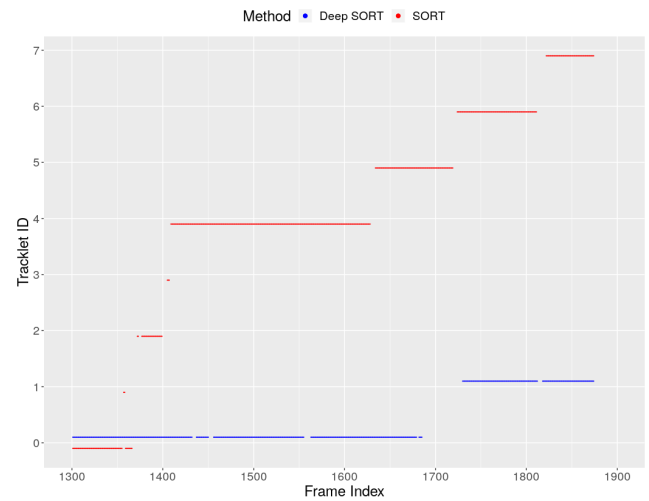


FIGURE 9. Representation of the detected tracklets for pig 1 from frame 1300 to 1875. This pig was visible for all frames, but showing the detail at this segment of frames was not possible if we showed all the tracklets from all frames. The Y-axis shows the tracklet IDs (which are independent for each method), the X-axis shows the frames during which the pig was visible. Red represents SORT generated tracklets, blue represents Deep SORT generated tracklets.

and a 30% decrease the number of fragmentations, which increases the average length of a track increases by 17.132 (+16%) and the average number of frames in which all pigs are perfectly tracked increases by 0.604 seconds (+11%). The maximum number of frames where all pigs are perfectly tracked, and the maximum frame length remains unchanged from those reported for SORT.

1) CASE STUDIES

In Figure 9 we show an example of Deep SORT's ability to consistently recover tracks between missed detections enables it to outperform SORT. The visualisation is of the tracklets generated for pig 1, the longest tracked pig in our dataset. The first tracklet in the visualisation, beginning at frame 1300, was lost and subsequently recovered 3 times when using Deep SORT. SORT is capable of recovery, but this is only possible when the detection is lost very briefly. This is why the gaps between recovered tracklets are consistently small, whereas Deep SORT can handle longer drops in detections, which is why Deep SORT is a much more robust method. This is also visible in Figure 10, which shows the detected tracklets for pig 12, the ID which benefited from the greatest reduction in IDSWs by using Deep SORT. SORT generated 18 tracklets for this identity, whereas Deep SORT only generated 7. We can see that tracklets generated under Deep SORT were much more stable than that of SORT; all whilst having having greater track coverage (Table 4).

D. BEHAVIOUR METRICS EXTRACTION RESULTS

The behavioural metrics extracted from from both SORT and Deep SORT tracks are shown in Table 5. Total distance and time spent idle scored well (0.010 MSE and 0.003 MSE respectively), however the average speed estimations

TABLE 5. Results of the behaviour extractions and the ground truth associated with them. Results are shown for SORT and Deep SORT. Distance is measured as the number of pixels travelled, average speed is measured as the average number of pixels travelled per second, and idle time is measured as the number of seconds a pig did not move more than 4 pixels. These results are normalised and the mean squared error (MSE) is shown for each (lower is better). The absolute error between the estimated behaviour and true behaviour for each method and metric is calculated; the number of IDs where this error is below a threshold is counted (higher is better).

ID	Distance			Avg. Speed			Idle Time		
	True	SORT	Deep SORT	True	SORT	Deep SORT	True	SORT	Deep SORT
1	7811	10868	12024	16	260	136	395	290	266
2	7368	6762	5830	53	50	75	79	42	52
3	3514	5638	5879	8	149	145	399	346	338
4	2224	3026	2814	9	12	11	226	214	213
5	1135	1498	2843	38	34	128	21	21	24
6	1745	1390	1050	56	94	81	18	10	10
7	397	579	520	25	35	34	12	5	5
8	264	1498	182	105	34	25	1	21	4
9	303	592	1256	11	21	60	22	18	26
10	179	366	316	10	21	19	11	10	10
11	1385	2791	3050	8	300	94	145	116	126
12	9749	5722	5346	63	798	217	103	47	155
13	1470	823	1619	90	132	219	8	5	40
14	994	1946	1841	7	15	14	115	89	88
15	1807	2922	2154	16	67	49	95	50	53
16	2720	2649	1733	25	172	172	91	51	40
17	229	1181	1174	1	9	9	116	109	108
18	2381	2090	1981	35	83	84	46	38	37
19	1393	1975	1755	6	32	26	191	181	178
20	2976	3916	3922	17	172	116	142	96	95
21	297	268	546	88	89	32	1	0	6
22	551	1713	1661	15	54	56	26	39	38
23	764	730	523	110	100	85	0	0	0
24	917	1125	1092	31	38	39	20	16	15
25	717	740	670	58	58	61	5	3	3
MSE ↓	-	0.010	0.015	-	0.148	0.008	-	0.003	0.008
Wilcoxon Test P-Value	-		0.221	-		0.037	-		0.331

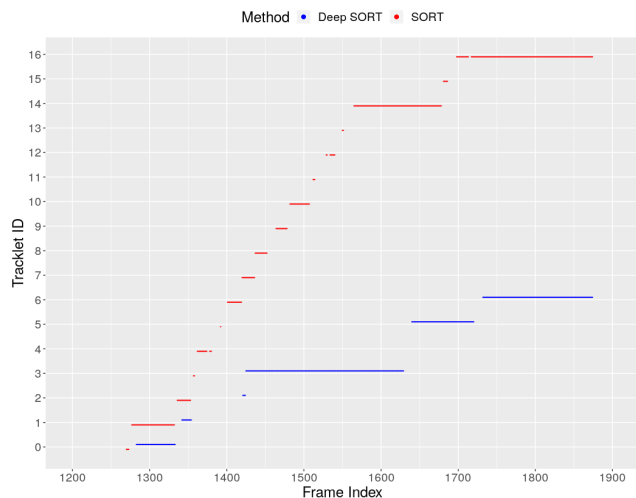


FIGURE 10. Representation of the detected tracklets for pig 12 for all the frames it was visible. The Y-axis shows the tracklet IDs (which are independent for each method), the X-axis shows the frames during which the pig was visible. Red represents SORT generated tracklets, blue represents Deep SORT generated tracklets.

substantially underscored (0.148 MSE). In particular, the average speed calculations were overestimated for the tracklets in almost all cases (80% of pigs). Estimations for pig 12 across all metrics derived from SORT were largely incorrect. This is reflected in Table 4, as this ID incurred a high number of IDSWs (16). The metrics derived from

Deep SORT, however, are more accurate for this pig due to the considerable decrease in IDSWs (5).

This relationship between a high number of IDSWs and poor estimations of behavioural metrics, specifically time spent idle and average speed, can be seen through all pigs (e.g. pigs 1 and 12). A higher number of IDSWs results in an overestimation of average speed. This is confirmed by the improved behaviour extraction when using Deep SORT (Table 5), which has a significantly lower error for average speed, which is why there is a substantial performance improvement for this metric when using Deep SORT (0.008 MSE, -95%). However, this relationship is much less consistent when calculating the total distance travelled. The total distance (0.015 MSE) and time spent idle (0.008 MSE) metrics extracted from Deep SORT show no statistically significant performance change over SORT.

IV. DISCUSSION

With the increasing use of deep learning for multi-object tracking [29], particularly in crowd analysis (i.e. people tracking), it is valuable to evaluate the extent to which these methods can be applied to more difficult applications. Applications can be more difficult for varying reasons such as: low FPS recordings, poor image quality, and similar looking objects.

Individual pig tracking on a commercial farm is an example of an application where all of these challenges can be found.

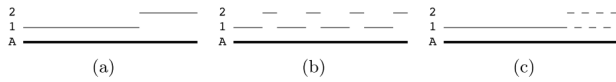


FIGURE 11. “Where there is one true identity A (thick line, with time in the horizontal direction), a tracker may mistakenly compute identities 1 and 2 (thin lines) broken into two fragments (a) or into eight (b, c). Identity 1 covers 67% of the true identity’s trajectory in (a) and (b), and 83% of it in (c). Current measures charge one fragmentation error to (a) and 7 to each of (b) and (c). Our proposed measure charges 33% of the length of A to each of (a) and (b), and 17% to (c).” (This figure and caption is from [26]).

This is mainly due to fact that the available infrastructure is limited and pigs are much less distinguishable from one another when compared to humans, especially with clothing taken into account. Deep learning methods for computer vision are much less common in the agricultural field, which tends to rely on classical signal processing methods, especially in the area of individual animal tracking. The ability to track individual pigs in real-time is key to creating a full system that can provide the information required for their management, including early detection of disease.

Previous research has focussed on how depth cameras can be utilised to track pigs under the challenging conditions presented by farm environments, such as poor image quality and low network bandwidth for data collection [3], [30]–[33]. These methods have been able to achieve good tracking results (89% MOTA [3]). However, they rely upon accurate depth sensor data which is only achievable at short distances, which limits the distance a camera can be placed from an object. This limits the field of view, and thus the pen coverages of a single depth-camera meaning more cameras are required to cover a large area when compared to the RGB cameras we use.

More recent research has applied deep learning models to the tracking of pigs [34] using object detection models, similar to the implementation used in our method, on RGB images, obtaining 89.58%. However, this depends upon IDs being sprayed on the pig and detecting each pig as a separate class. This is not a feasible or practical in a commercial setting due to the large number of pigs that may reside within a pen. Moreover, the sprayed markers do not hold for long and therefore would need to be continually reapplied for reliable tracking. Secondly, the model needs to be specifically trained on each numerical ID that it needs to track, which impedes the generalisability of the method.

You Only Look Once (YOLO) [35] is a commonly used method for object detection that is popular for its fast inference times was considered for use in our application. However, despite the real-time inference, its performance on detecting smaller objects is much poorer than that of Faster R-CNN. YOLOv3 does attempt to rectify this, but it does so at the cost of poorer performance on larger objects relative to Faster R-CNN [36]. As we wanted our method to generalise to different environments such as taller buildings where the camera is mounted higher, thus making pigs smaller in the images, and because we do not deal with high FPS images, we decided against using YOLO.

The use of standardised metrics for MOT in all applications is key to summarising how well methods perform in various domains and applications as concisely and as accurately as possible. Of the research discussed above, several report MOTA, but none report IDF1. This poses a risk of skewing the effectiveness of the tracking method that has been implemented, by misrepresenting the effect of IDSWs within the system. Figure 11 (taken from [26]) demonstrates an example where there are varying examples of IDSW. Based on these tracklets, MOTA would favour (a), as it has the fewest number of fragmentations; where IDF1 would favour (c) it provides a greater identity coverage (83% vs 67%). Hence, we believe that in our context, the under-reported IDF1 metric is a more informative tracking quality metric than the more widely used MOTA.

Our method was able to detect, track and extract behavioural metrics of individual pigs in real-time using a Faster R-CNN for localisation and Deep SORT for tracking. Alternative methods for tracking do exist (e.g. a kernelized correlation filter [37]), however, we chose SORT and Deep SORT as they provide a good side-by-side comparison of trajectory-based tracking and a combined trajectory & visual-based tracker respectively. This results in our main contribution, which is a complete, end-to-end system that can process raw images and produce behavioural metrics for individual pigs, whilst maintaining the identity of pigs between frames. We relied solely on footage recorded from an inexpensive RGB camera, as opposed to expensive 3D depth cameras, recording at an average of only 4 FPS and a resolution of 1920×1080 . This low FPS was a limitation of the hardware that was used, as it was responsible for other background tasks which limited the number of frames that could be captured.

From this data we were able to achieve comparable performance to previous research in terms of MOTA (95%), with the improvement that our method can do so without the use of additional or expensive hardware, such as RFID tags and depth cameras, or visual aids, such as painted IDs. We were also able to use this data to reliably re-identify pigs when they become occluded or the detector fails to localise them (0.862 mAP), which enables us to achieve an IDF1 of 73.4%. This use of visual appearance when assigning identities at runtime proved valuable, as the method was able to more accurately determine the average speed of a pig, without compromising on other metrics, as this is easier to do when the number of IDSWs is lower. As our data was restricted to the pig sizes that were available, we were not able to verify whether the method will work as pigs grow.

Where other research focussed on performance aspects (e.g. weight) of individual pigs [38], we looked specifically at how active pigs are by extracting movement related behavioural metrics from the generated tracks. We recorded pig behaviours for up to a theoretical maximum of 10 minute video segments. This is equivalent to the behavioural method of scan sampling, where all of the actions of all animals are recorded for intervals in order to obtain behavioural metrics

for individuals within a group [39]. We have shown not only that we are able to successfully localise and track pigs from challenging naturalistic settings (commercial farms), but also that we are able to successfully extract a range of domain-relevant, useful knowledge from the outputs of fairly generic object localisation and tracking algorithms as the ones that we have used. One of the advantages of the method is that we are able to extract several behaviours from the same pig in real-time. This is beneficial, as it is suggested that a combination of behavioural metrics might be a better indicators of pig health than a change in a single behaviour [40].

V. CONCLUSION

We have implemented a system to detect and track pigs in a commercial farm setting using deep learning that allows us to track pigs for up to 4 minutes, with an MOTA of 92% and IDF1 of 73.4% without the use of additional hardware or visual aids. The tracks derived from this system are able to be used to calculate behavioural metrics for total distance travelled, average speed and time spent idle for individual pigs. The length of the identified tracks is mostly limited by the length of the realistic video we could use to evaluate the method, due to technical constraints in the data capturing system. However, this method generates a set of tracklets that (mostly) successfully cover parts of the annotated tracks and in cases where detections are missed, is capable of recovering the identity. Overall, our work shows how deep learning algorithms enable the development of a relatively cheap (as we just require a standard RGB camera) pig monitoring system that can provide useful information to characterise pig's behaviour by applying transfer learning strategies on top of a (by now) standard object localisation method such as Faster R-CNN. The literature shows how such descriptors enable the creation of more personalised pig treatment plans [1], [2] which in turn decrease disease risk and reduce use of medication, whilst maintaining animal performance.

ACKNOWLEDGEMENTS

This work was conducted under the Feed-a-Gene project.

REFERENCES

- [1] S. G. Matthews, A. L. Miller, J. Clapp, T. Plötz, and I. Kyriazakis, "Early detection of health and welfare compromises through automated detection of behavioural changes in pigs," *Vet. J.*, vol. 217, pp. 43–51, 2016.
- [2] S. H. Richter and S. Hintze, "From the individual to the population- and back again? Emphasising the role of the individual in animal welfare science," *Appl. Animal Behav. Sci.*, vol. 212, pp. 1–8, Mar. 2018.
- [3] S. G. Matthews, A. L. Miller, T. Plötz, and I. Kyriazakis, "Automated tracking to measure behavioural changes in pigs for health and welfare monitoring," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 17582.
- [4] Z. Berk, Y. C. Laurenson, A. B. Forbes, and I. Kyriazakis, "Modelling the consequences of targeted selective treatment strategies on performance and emergence of anthelmintic resistance amongst grazing calves," *Int. J. Parasitol., Drugs Drug Resistance*, vol. 6, no. 3, pp. 258–271, 2016.
- [5] A. Yigit and A. Temizel, "Individual and group tracking with the evaluation of social interactions," *IET Comput. Vis.*, vol. 11, no. 3, pp. 255–263, 2016.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.
- [8] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
- [9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [10] M. Everingham, L. V. Gool, K. I. Christopher Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. Accessed: Aug. 1, 2019. [Online]. Available: <http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html>
- [11] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 868–884.
- [12] M. Ju, Y. Choi, J. Seo, J. Sa, S. Lee, Y. Chung, and D. Park, "A Kinect-based segmentation of touching-pigs for real-time monitoring," *Sensors*, vol. 18, no. 6, p. 1746, 2018.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [14] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, D, J. Basic Eng.*, vol. 82, pp. 35–45, 1960.
- [15] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
- [16] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 748–756.
- [17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [19] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
- [20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
- [21] S. H. Rezaatofghi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic matching using m-best solutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 136–145.
- [22] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1846–1855.
- [23] X. Lan, X. Zhu, and S. Gong, "Person search by multi-scale matching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 536–552.
- [24] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*. [Online]. Available: <https://arxiv.org/abs/1603.00831>
- [25] C. Heindl, (2019). *PY-Motmetrics*. [Online]. Available: <https://github.com/cheind/py-motmetrics>
- [26] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 17–35.
- [27] E. A. Gehan, "A generalized wilcoxon test for comparing arbitrarily singly-censored samples," *Biometrika*, vol. 52, nos. 1–2, pp. 203–224, 1965.
- [28] K. Bernardin, A. Elbs, and R. Stiefelhof, "Multiple object tracking performance metrics and evaluation in a smart room environment," in *Proc. 6th IEEE Int. Workshop Vis. Surveill., Conjoint ECCV*, vol. 90, 2006, p. 91.
- [29] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," *Pattern Recognit. Lett.*, vol. 107, pp. 3–16, May 2018.
- [30] M. Mittek, E. T. Psota, J. D. Carlson, L. C. Pérez, T. Schmidt, and B. Mote, "Tracking of group-housed pigs using multi-ellipsoid expectation maximisation," *IET Comput. Vis.*, vol. 12, no. 2, pp. 121–128, 2017.

- [31] J. Kim, Y. Chung, Y. Choi, J. Sa, H. Kim, Y. Chung, D. Park, and H. Kim, "Depth-based detection of standing-pigs in moving noise environments," *Sensors*, vol. 17, no. 12, p. 2757, 2017.
- [32] J. Lee, L. Jin, D. Park, and Y. Chung, "Automatic recognition of aggressive behavior in pigs using a Kinect depth sensor," *Sensors*, vol. 16, no. 5, p. 631, Mar. 2016.
- [33] J. Sa, Y. Choi, H. Lee, Y. Chung, D. Park, and J. Cho, "Fast pig detection with a top-view camera under various illumination conditions," *Symmetry*, vol. 11, no. 2, p. 266, 2019.
- [34] Q. Yang, D. Xiao, and S. Lin, "Feeding behavior recognition for group-housed pigs with the faster R-CNN," *Comput. Electron. Agricult.*, vol. 155, pp. 453–460, Dec. 2018.
- [35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
- [36] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <https://arxiv.org/abs/1804.02767>
- [37] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [38] M. Mittek, E. T. Psota, L. C. Pérez, T. Schmidt, and B. Mote, "Health monitoring of group-housed pigs using depth-enabled multi-object tracking," in *Proc. Int. Conf. Pattern Recognit., Workshop Vis. Observ. Anal. Vertebrate Insect Behav.*, 2016, pp. 1–4.
- [39] P. Martin, P. P. G. Bateson, and P. Bateson, *Measuring Behaviour: An Introductory Guide*. Cambridge, U.K.: Cambridge Univ. Press, 1993.
- [40] A. L. Miller, H. A. Dalton, T. Kanellos, and I. Kyriazakis, "How many pigs within a group need to be sick to lead to a diagnostic change in the group's behavior?" *J. Animal Sci.*, vol. 97, no. 5, pp. 1956–1966, 2019.



JAKE COWTON received the B.Sc. degree in computer science from Northumbria University, in 2015. He is currently pursuing a Ph.D. degree with Newcastle University. His current research interests include deep learning and computer vision.



ILIAS KYRIAZAKIS is currently a Professor of animal health and nutrition with the School of Natural and Environmental Sciences, Newcastle University, U.K. He is also a Veterinarian by training, who is interested in the consequences of management on the health and performance of livestock. More recently, he has been involved in the introduction of disruptive technologies in livestock systems, which include the application of sensors to monitor remotely the health, welfare, and productivity of animals. This paper falls within this research area, as it aims to develop an early warning system for disruption in health and welfare of pigs, through the automated monitoring of their behaviour.



JAUME BACARDIT received the B.Eng., M.Eng. degrees in computer engineering, and the Ph.D. degree in computer science from Ramon Llull University, Spain, in 1998, 2000, and 2004, respectively. He is currently a Reader in machine learning with Newcastle University, U.K. His research interests include the development of machine learning methods for large-scale problems, the design of techniques to extract knowledge and improve the interpretability of machine learning algorithms, and the application of these methods to a broad range of problems, mostly in biomedical domains.

...