Automatically detecting and tracking goat position by 2D camera imaging and deep learning

Djahlin Nikue Amassah*, Xavier Desquesnes*, Bruno Emile*, and Sylvie Treuillet*
* Université d'Orléans - PRISME laboratory
Email: djahlin.nikue-amassah@etu.univ-orleans.fr

***Abstract*—This work is performed within the project AniMov, which consists in building a video surveillance system of animal behaviors in a livestock situation. The main objective of the project is to provide farmers with an analysis tool capable of producing precise indicators to control feeding and reproduction but also to detect activity cycles and abnormal situations. The work presented in this article is the first part of this project: the detection and tracking of goats, which allows us to track the general activity of the livestock. For the detection, we used YOLO v4, a one-step detection architecture, after a comparison with the Faster R-CNN model. For the tracking, we implemented and compared SORT and Deep SORT algorithms. The evaluation of our detection method gives an average accuracy of 86.74% and 90.56% respectively for classes "standing_goat" and "lying_goat". For tracking, we obtained an average association accuracy of 72% with SORT and 74% with Deep SORT.**

I. Introduction

The AniMov project aims to provide farmers with an automatic analysis tool for their livestock using a real-time video surveillance system. Farmers want to know the behavior of the animals such as feeding, watering, reproduction and parturition, in order to have precise indicators to control the feeding and the reproduction of the livestock. The permanent and direct observation of the animals by a human is not possible due to economic reasons and also of alteration of the animals' behavior. We build a vision system to automatically analyze animals' postures (standing and lying) and track their activity cycles. The work presented in this article focuses on the detection and tracking of goats. The detection and tracking of multiple objects remains a great challenge in the field of computer vision. This task is particularly complex when it comes to tracking animals in enclosures like goats livestock (figure 2). The high density of goats in the pens increases the number of occlusions and can lead to detection failures. The creation of training and test data is also very costly due to the lack of an existing dataset in this context.

II. Related Work

Over the past two decades, researchers have investigated a variety of video camera-based methods and technologies for detecting and tracking animals in livestock situations.
Using traditional color imaging and background subtraction, the researchers designed methods for tracking in constrained environments where pigs walk individually in front of the camera [1]. The classical methods of tracking multiple objects show limitations in the context of a herd of very similar individuals as it is the case in a livestock situation. To monitor several animals simultaneously, it is necessary to segment them both from the background and from each other; a difficult task given their tendency to cluster. Kashiha et al. [2] proposed an automated method to identify marked pigs in a pen using pattern recognition techniques. First, a segmentation is performed using a 2D Gaussian filter (to denoise the image) followed by Otsu thresholding. Next, the marks on the pigs were extracted using a similar segmentation method, which was then used to identify the pattern based on a Fourier description. It is also very sensitive to the lighting changes. For this purpose, depth information was introduced into the tracking of the animals using depth cameras: multiple tracking in 3D.

For tracking by 3D cameras, the authors in [3] applied the kinect v2 depth camera for the monitoring of pigs. The upright camera requires manual calibration of the system where the user selects corner points defining the boundaries of the pen, feeder, waterer, heat mat and the position of each pig. 3D point clouds and ellipsoid tracking are used with a Kalman filter to estimate the position and orientation of each pig. Other researchers have also proposed a 3D tracking system using the region growth algorithm. Pigs were tracked by linking detections in consecutive images. The Hungarian algorithm, as described in [4], was used for the association between the images by performing a combinatorial optimization of all pig to pig assignments. Although existing vision systems based on depth video cameras have achieved some success in detecting and tracking livestock, they have some drawbacks. The Kinect depth camera has a limited range of 4 meters and a limited field of view (horizontal 58.5 degrees and vertical 45.6 degrees). In addition, the accuracy of the depth data is very sensitive to the position of the camera [5].

In [5], the authors have implemented a 2D color camera based pig detection and tracking software. They compared the R-FCN, Faster R-CNN and SSD object detection architectures. The Faster R-CNN and R-FCN have shown good detection accuracies, but they are slower than the SSD. The SSD architecture was chosen for detection. Then, for tracking, the Discriminative Correlation Filters (DCF) is used with

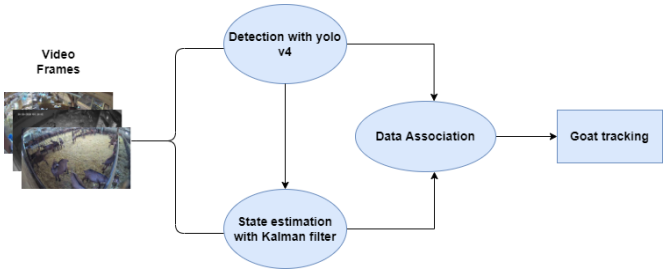

Fig. 1. Proposed method for detection and tracking of goats in a breeding farm

Fig. 2. dataset images

the Hungarian algorithm for data association. J. Cowton et al. [6] proposed an automated individual pig tracking method using the Faster R-CNN. For tracking, they evaluated two methods: SORT and Deep SORT. The SORT method combines the Kalman filter and the Hungarian algorithm for tracking. This method does not require any training and can be directly applied on the output of the detection. In addition to the Kalman filter and the Hungarian algorithm, the deep sort uses a learned association metric to determine if two consecutive images contain the same object. Unlike SORT, Deep SORT is less dependent on the accuracy of detections, although it still requires them to be of good quality, as it still partially uses the Kalman filter to make association decision [6]. Their tracking method was evaluated using the Multi-Object Tracking Accuracy (MOTA) [7] metric. They obtained 95.1 % of MOTA with SORT and 92.1 % with Deep SORT. Other researchers have used an R-CNN and LSTM-based architecture for cow tracking [8]. The detection was performed with the two-step Faster R-CNN detection architecture and a Long-term Recurrent Convolutional Networks (LRCN) architecture for tracking. In the LRCN, the visual features of the input video images are extracted by a CNN to be fed into an LSTM layer that finally produces an identity prediction.

III. Method

A. Dataset and Annotations

To perform goat detection, we built a dataset of training and test images. We used the video frames from the 2D RGB cameras placed in the goat pens. The videos were retrieved from two different goat farms. The high resolution 1920x1080 pixels RGB cameras are placed in each corner of the enclosure to have different viewpoints. In the enclosure, we can have up to 4 cameras. For the training of our detection model, we used the images from all the cameras. But for the inference phase, we process images from only one camera at a time. All images are were taken in full RGB. Figure 2 shows some examples of the images in our dataset.We built a set of 796 day and night images (they are treated in the same way) : 646 for training and 150 for test. The training and test data contains examples of all the challenging scenarios encountered. We annotated the images using the labelImg[1] tool. We considered 2 classes: "standing_goat" and "lying_goat". The dataset is built in such a way as to balance the proportion class in each image. The annotation consists of drawing bounding boxes around each goat in the images and generating a text file per image containing the coordinates of the bounding boxes. These files are used as ground truth (GT) to train our detection model. We also have other classes like: feeding, waterer and scrubbing but they are not integrated in our annotated dataset. These classes are detected using the positions of the feeder, the drinker and the scrubber in the pen compared to the goats positions.

B. Network architecture for detection

For detection we trained and compared two architectures: the Faster R-CNN and YOLO v4. Faster R-CNN is one of the best 2-step detection models with an architecture that integrates a region proposal network (RPN) to generate initial regions of interest (RoI) for subsequent learning. This network architecture gives good accuracy but remains very slow for a real time system [9]. In our system, as shown in the figure 1, the detection is performed in each image by the YOLO v4 [11] architecture which is more adapted for a real time system. It is a convolutional neural network architecture allowing object detection in a single step, which makes it very fast compared to 2-step detection architectures. Most modern sensing models require multiple GPUs for training with a large mini-batch size, and doing so with a single GPU makes training very slow and impractical. YOLO v4 solves this problem by creating an object detector that can be trained on a single GPU with a smaller mini-batch size. To train our detection network architecture we used a batch size of 64, subdivision 16, a learning rate of 0.001, a momentum of 0.9, decay 0.005 and 1000 epochs.

C. Tracking

We have implemented and compared SORT and Deep SORT methods for tracking. For the SORT method, we have implemented and tested Kalman and particle filter. For the final system, we used the Kalman filter as state estimation because it gives us a better speed compared to the particle filter. Opposed

[1]https://github.com/tzutalin/labelImg

to the Kalman filter [12] the particle filter can model non-linear object motion because the motion model should not be written as a state transition matrix like in the Discrete Kalman filter. Moreover, the particle filter [13] is fairly easy to understand, but there is a disadvantage: the performance of the filter depends on the particles number where a higher number of particles leads to a better estimate, but it is more costly. For the Kalman filter, mathematical functions are used to detect the state mean and covariance. In our case the state vector E_t is modeled as follows : $E_t(x_t, y_t, v_{xt}, v_{yt}, w_t, h_t)$ where x_t and y_t represent the coordinates of the center of the bounding box of the object, v_{xt} and v_{yt} its speed; and w_t, h_t its width and height at time t. For the particle filter, in state estimation, the particles are generated in each bounding box predicted by YOLO v4. Each particle incorporates tests if or not it is likely that the object is at the position where the particle is. After the particles have been evaluated, the weights are assigned according to how good the particles are. Then the good particles are multiplied and the bad particles are removed through the re-sampling process. The next particle generation predicts where the object might be. This generation is evaluated, and the cycle repeats. The association between the box (P) predicted by yolov4 and the one estimated by the filter $(P^{'})$ was done by Euclidean distance calculation : $d(P, P^{'}) = \sqrt{(P - P')^2}$ where $P = (x, y, w, h)$ and $P^{'} = (x^{'}, y^{'}, w^{'}, h^{'})$. x, y are the center of the box and w, h are the width and height.

For the Deep SORT method, we trained a deep association metric model. We used our previously implemented SORT method to automatically generate the tracklets database for the deep association metric model. In this database we have 229 IDs which correspond to different goats. We used 9 videos sequences, from 4 cameras, to generate the whole training database. Along with the Kalman filter and Hungarian algorithm used in SORT, Deep SORT uses the learned association metric to determine whether or not two images of a goat contain the same goat. This association metric is learned using a deep learning CNN model which uses a reparametrisation 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 [15].

IV. Experiments and Basic Results

A. Detection

The training and testing was performed using an Intel(R) Core(TM) i9-9900K CPU at 3.60GHz, 2 Nvidia Titan RTX GPUs and 64GB of DDR3 RAM. The mean value of average accuracy (mAP), PR curve (figure 4) and frames per second (FPS) are the metrics used to evaluate the two object detection architectures (YOLO v4 and Faster R-CNN) tested. A prediction is considered true (TP: True Positive) if the IoU is above a given threshold (0.5 in our case), and false (FP: False Positive) if it is below. The undetected ground truth corresponds to False Negative (FN). The calculation of precision, recall and mAP can be found in [16]. Figure 3 shows us the output of the detection on test image.

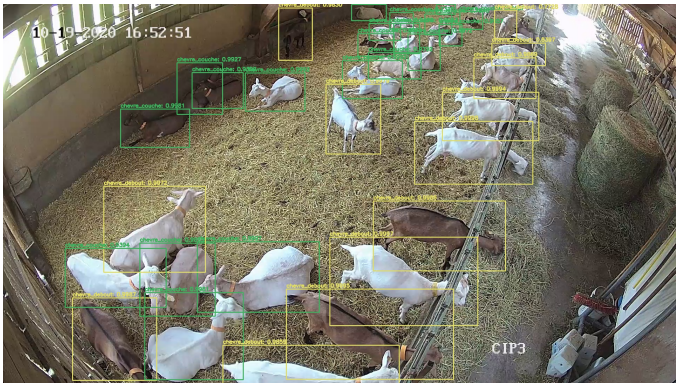

Fig. 3. Detection output with YOLO v4

TABLE I
Comparison of YOLO v4 and Faster R-CNN
test images number : 150

field	*YOLO v4*	*Faster R-CNN*
mAP@.5	88.65%	72.41%
standing_goat AP@.5	86.74%	68.69%
lying_goat AP@.5	90.56%	76.12%
FPS	95.3	25.5

Figure 5 shows a graph generated from our system allowing us to track the general activity of the livestock over a period of 6 hours (from 12 to 5 pm). Farmers wanted to have an overview of the number of animals lying down, standing, feeding or drinking at different times of the day. This information will allow to know the resting time of the herd and their feeding time. These times can also be obtained for an individual goat through tracking.

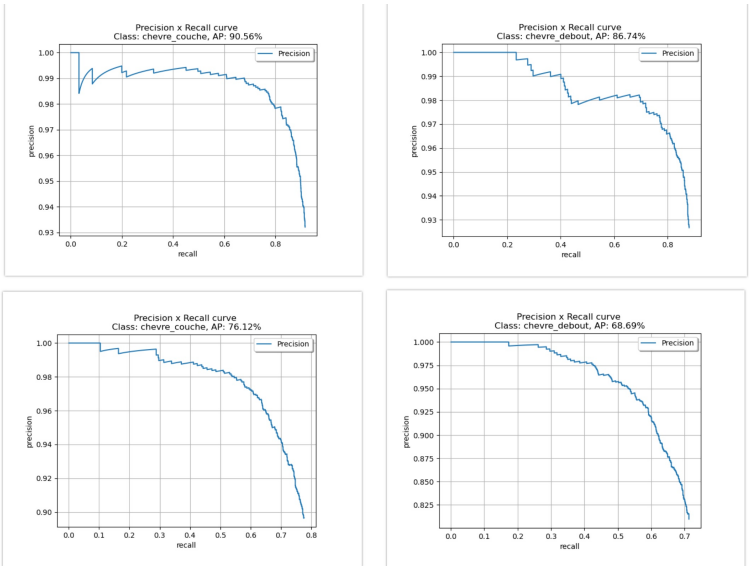

Fig. 4. PR Curve of YOLO v4 (top) and Faster R-CNN (bottom)

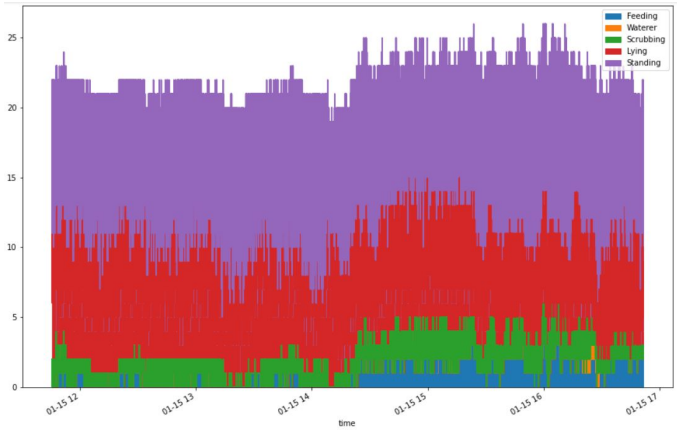

Fig. 5. Livestock monitoring

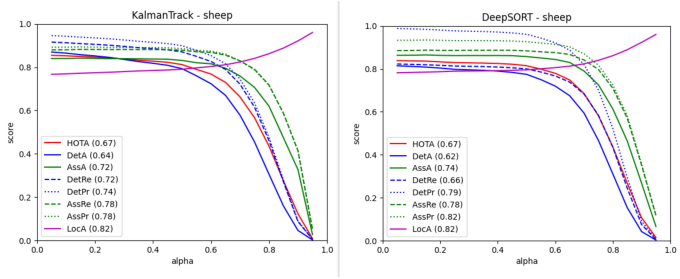

Fig. 6. Tracking Evaluation (SORT (left), Deep SORT (right))

B. Tracking

We evaluated and compared our tracking methods (SORT and Deep SORT) using the HOTA metric [14]. With HOTA we can evaluate different aspects of tracking separately compared to existing metrics such as MOTA and MOTP which focus more on detection rather than on association. This enables clear understanding of the different types of errors that trackers are making and enables trackers to be tuned for different requirements.

For the tracking evaluation and the HOTA metric computing, we used 4 sequences of videos about 1 minute.

As can be seen in figure 6, we have the values of the accuracies and recalls of each metric to calculate HOTA (alpha is IoU treshold). The equations for the calculation of these metrics can be found in [14]. With these values we can evaluate our tracking method on several levels such as data association, detection or localization. In our case, for example, we have an average association accuracy of 74% for Deep SORT and 72% for SORT (figure 6), which measures how well our tracking method links detections over time into the same identities (IDs).

V. Conclusion

In this work, we present our method for automatically detecting and tracking goat position by 2D camera imaging and deep learning in a challenging environment. After a comparison between YOLO v4 and the Faster R-CNN, we selected YOLO v4, a one-step detection architecture that, produces better accuracy and faster detection time. The evaluation of our detection method gives us an average accuracy of 86.74% for the ”standing_goat” class and 90.56% for the ”lying_goat” class. For tracking we obtained an average association accuracy of 72% with the SORT and 74% for Deep SORT. In future work, we will implement an LSTM-based architecture to recognize some behaviors of goats and also perform tracking at the same time. We will therefore have an end-to-end architecture allowing the detection, tracking and recognition of goat behavior.

Acknowledgment

This work is part of the FC9513/APR IR 2019 project AniMov Animal Movements Observation (AniMov), supported by the Région Centre-Val de Loire (France). The authors would like to thank the ”Conseil Régional du Centre - Val de Loire” and all the partners of this project.

References

[1] N. M. Lind, M. Vinther, R. P. Hemmingsen and A. K. Hansen, *Validation of a digital video tracking system for recording pig locomotor behaviour*, Journal of neuroscience methods, Vol. 143(2), pp. 123–132, 2005.

[2] L. Zhang, H. Gray, X. Ye, L. Collins, N. Allinson, *Automatic Individual Pig Detection and Tracking in Pig Farms* Sensors (Basel) 19-01188, pp. 1-20, 2019.

[3] M. Mittek, E. T. Psota, L. C. Pérez, T. Schmidt, B. Mote, *Health Monitoring of Group-Housed Pigs using Depth-Enabled Multi-Object Tracking*, In Proceedings of International Conference Pattern Recognition, Workshop on Visual observation and analysis of Vertebrate And Insect Behavior, pp. 4, 2016.

[4] H.W. Kuhn, *he hungarian method for the assignment problem*, Naval research logistics quarterly, Vol. 2, pp. 83-97, 1955.

[5] N.J.B. McFarlane, C.P. Schofield, *Segmentation and tracking of piglets in images*, Machine Vision and Applications, Vol. 8, pp. 187-193, 1995.

[6] J. Cowton I. Kyriazakis J. Bacardit, *Automated Individual Pig Localisation, Tracking and Behaviour Metric Extraction Using Deep Learning*, IEEE Access, Vol. 7, pp. 108049–108060, 2019.

[7] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K.Schindler,*MOT16: A Benchmark for Multi-Object Tracking*, Arxiv, pp. 1-12, 2016.

[8] W. Andrew, C. Greatwood, T. Burghardt, *Visual Localisation and Individual Identification of Holstein Friesian Cattle via Deep Learning*, In Proceedings of the IEEE International Conference on Computer Vision, pp.2850-2859, 2017.

[9] S. Ren, K. He, R. Girshick, J. Sun, *Faster R-CNN: Towards Real-Time ObjectDetection with Region Proposal Networks*, Arxiv, pp. 1-14, 2016.

[10] F. Zeng, B. Dong, T. Wang, C. Chen, X. Zhang, Y. Wei, *MOTR: End-to-End Multiple-Object Tracking with TRansformer*, Arxiv, pp. 1-10, 2021.

[11] A. Bochkovskiy, C. Wang, H. M. Liao, *YOLOv4: Optimal Speed and Accuracy of Object Detection*, Arxiv, pp. 1-17, 2020.

[12] M. Marrón, J.C. García, M.A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, J. Cerro, *Comparing a Kalman Filter and a Particle Filter in a Multiple Objects Tracking Application*, In IEEE International Symposium on Intelligent Signal Processing, pp. 1-6, 2007.

[13] M. Jaward, L. Mihaylova, N. Canagarajah and D. Bull, *Multiple Object Tracking Using Particle Filters*, IEEE Aerospace Conference, pp. 1-8, 2006.

[14] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, Laura Leal-Taixé and B. Leibe1, *HOTA: A Higher Order Metric for Evaluating Multi-object Tracking* , International Journal of Computer Vision, pp. 548-578, 2021.

[15] N. Wojke and A. Bewley, *Deep cosine metric learning for person reidentification* in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), pp. 748–756, 2018.

[16] R. Padilla, S. L. Netto and E. A. B. da Silva, *A Survey on Performance Metrics for Object-Detection Algorithms* in International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237–242, 2020.