Analysis of Dogs' Sleep Patterns using Convolutional Neural Networks

Anna Zamansky[1], Aleksandr M. Sinitca[2], Dmitry I. Kaplun[2], Michael Plazner[1], Ivana G. Schork[3], Robert J. Young[3], and Cristiano S. de Azevedo[4]

[1] Information Systems Department, University of Haifa, Israel
[2] Saint Petersburg Electrotechnical University "LETI", Russia
[3] School of Environmental and Life Sciences, University of Salford, UK
[4] Department of Biodiversity, Evolution and Environment, Federal University of Ouro Preto, Brazil

Abstract. Video-based analysis is one of the most important tools of animal behavior and animal welfare scientists. While automatic analysis systems exist for many species, this problem has not yet been adequately addressed for one of the most studied species in animal science — dogs. In this paper we describe a system developed for analyzing sleeping patterns of kenneled dogs, which may serve as indicator of their welfare. The system combines convolutional neural networks with classical data processing methods, and works with very low quality video from cameras installed in dogs shelters.

Keywords: convolutional neural networks, animal science, animal welfare, computer vision

1 Introduction

Video-based analysis is one of the most important tools of animal behavior and animal welfare scientists. For instance, it is very useful for measuring *time budget* of animals, a common ethological and welfare parameter, indicating the amount or proportion of time that animals spend in different behaviors [1]. In this case the data to be analyzed may amount of hundreds of hours of data, and is a tedious and error-prone task. Naturally, automatic video analysis has the potential to revolutionize the work of animal scientists in terms of precision, nature and number of behavioral variables that can be measured, and volumes of video data that can be processed. Automatic video-based systems already exist for different species: wild animals [2], pigs [3,4], poultry [5], insects [6], and many more. Moreover, well-developed commercial systems for rodent tracking such as Ethovision [7,8] are widely used in behavioral research.

Dogs are a widely studied species in animal science. While video analysis is widely applied in the context of dogs (see, e.g. [9,10]), very few works address *automatic* video-based analysis of dog behavior [11,12,13]. All of these works use video from 3D Kinect camera, the installation and use of which is not trivial and also quite expensive.

Our approach takes a different strategy, using the simplest web or security cameras footage, and paying a "computational" price instead for the system's learning. It is a part of our ongoing multi-disciplinary project for automatic analysis of dog behavior, based on video footage obtained from simple cameras (an overview of the project can be found in [14]; preliminary ideas were presented in [15]). In this paper we present a system developed for supporting an ongoing research project in animal science, investigating sleeping patterns of kennelled dogs as indicators of their welfare. Our system was developed for automatically quantifying dogs' sleeping patterns. It combines convolutional neural networks with classical data processing methods; it works with very low quality video data, and supports detecting multiple dogs in a frame. In what follows we describe in further details the research problem and the developed solution.

2 Related Work

Automatic tracking and behavior analyzing systems are used for wild animals [2], pigs [3,4], poultry [5], insects [6], and many more. Well-developed systems for rodent behavior recognition such as Ethovision [7,8] are widely used in behavioral research. In the context of dogs, automatic quantification of animal activities have mostly been explored in relation to pet wearables. These include a plethora of commercially available canine activity trackers (such as FitBark[5], Whistle[6] or PetPace[7]). While such devices can measure activity and sleep patterns, none of them has yet been scientifically validated, and thus are not always appropriate to be used in clinical and scientific settings. Wearables have been investigated in the context of predicting the success of future guide dogs ([16,13]), impacting the bonding between dog and owner[17,18], and supporting the relationship between guide dog centers and puppy raisers ([19]). van der Linden et al. [20] provide a comprehensive overview of commercially available dog trackers, discussing also their privacy implications. yet ripe to be used for scientific research or clinical settings. Fair accuracy was achieved for several self-developed sensor-based activity trackers [21,22,23], which are limited to a small number of basic positions and postures.

Barnard et al. addressed a similar problem of automatic behavioral analysis of kennelled dogs using 3D video monitoring [11]. Dog body part detection was done using standard Structural Support Vector Machine classifiers, and automatic tracking of the dog was also implemented. However, as discussed in the introduciton, this approach requires expensive equipment and non-trivial installation of 3D cameras (such as Microsoft Kinect). Our approach, on the other hand, can use video footage of very low quality, obtained from simple, cheap and easily available cameras.

[5] See: https://www.fitbark.com/

[6] See: https://www.whistle.com/

[7] See: https://petpace.com/

3 Problem Definition

The above mentioned animal science study[8] is a collaboration between the University of Salford and the Animal Science Center of Universidade Federal de Ouro Preto in Brazil. It focuses on analyzing sleep patterns of breeding stock kenneled dogs as welfare indicators. The dogs, bred and maintained by the Animal Science Center in Brazil, were captured for eight consecutive months using simple security cameras installed in their kennels (using night vision at night). The collected video data is of size 2.1 TB and contains 13,668 videos, comprising over 4,000 hours of footage. Each of the kennel rooms house either one or two dogs. The cameras are able to capture videos in two modes: full-color mode, where the space is illuminated by the sun or a lamp, and gray-scale mode, where the space is illuminated by infrared camera light. Despite their HD resolution (1280x720), the video footage is of a very low quality.

The main problem consists of automatically computing the following sleep parameters for each dog, which have been recognized as important in the study:

- total amount of sleep – the number of frames in the video where the dog is asleep (i.e., lying down, eyes closed);
- sleep interval count – the number of blocks of consecutive frames where the dog is asleep in every frame;
- sleep interval length – the number of frames in a given sleep interval.

Our aim is to automatically compute these parameters for each dog by (i) localization of the dog in each frame, and (ii) classification of its state as awake or asleep. We henceforth focus on these two tasks and evaluate the performance of the system in relation to the final task (ii).

4 System Description

An overview of the system's client/server architecture is provided in Fig. 1. The input to the system is a video, and its output is a summary of the sleep parameters for that video. The video is processed by the client, and sent to the server frame by frame. The frames serve as input to a neural network, which has two main tasks: marking the dog's position, and classifying the sleep/awake state of each dog that was identified. The images are fed to the model in a sequence, which the network processes one-by-one without keeping state.

In what follows, we describe in further detail the dataset used to train the neural network, our experiments with the networks' different architectures, the post-processing methods applied to correct the network's outputs, and, finally, the calculation of the sleep parameters.

[8] The study was approved by the ethical panels of both institutions; protocol numbers: University of Salford Ethical Approval Panel - STR1617-80, CEUA/UFOP (Brazil) - 2017/04

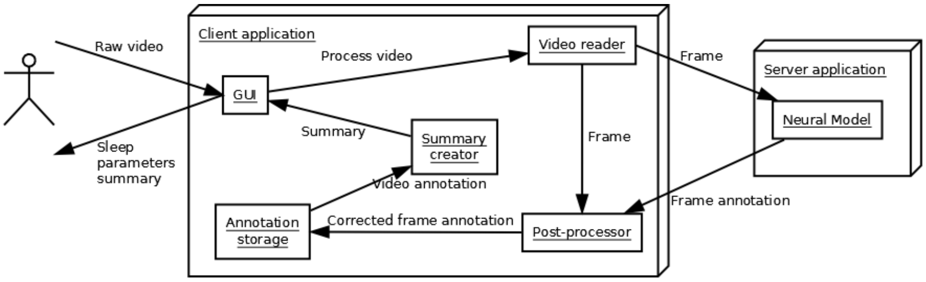

Fig. 1. System Architecture.

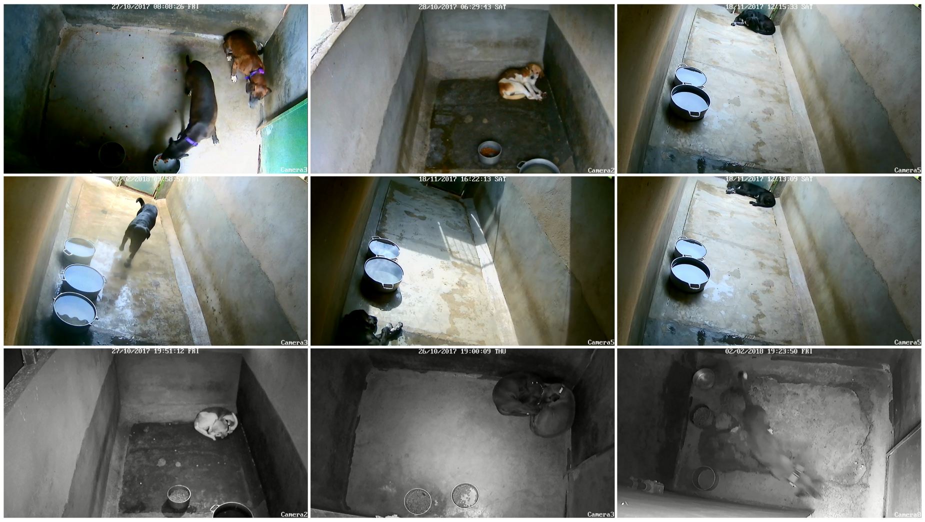

Fig. 2. Example of frames.

4.1 The Dataset

Our training dataset consisted of 8000 frames extracted from the videos (see Fig.

The obtained frame annotations included two attributes for every dog visible in the frame:(i) bounding box: an axis-aligned box surrounding each identified dog, and (ii) state of the identified dog: awake or asleep.

The annotation was performed by the first three authors independently, reaching a consensus via discussion in controversial situations (e.g., when the dog's eyes are not visible), and consulting with the last three authors who are animal experts. Frames where the dog was not clearly seen or hidden behind objects were discarded.

4.2 The Neural Network

The neural network has two tasks: (i) localization, i.e., marking the dog's position with a bounding box, and (ii) classification, i.e., marking sleep/awake state of each dog that was identified. To this end we considered two possibilities:

- Two-stage model: two distinct neural networks for the two tasks of localization and classification, packaged as one model (see Fig. 3).
- One-stage model: an end-to-end model for the detection (both localization and classification) of two types of objects: a sleeping and an awake dog.

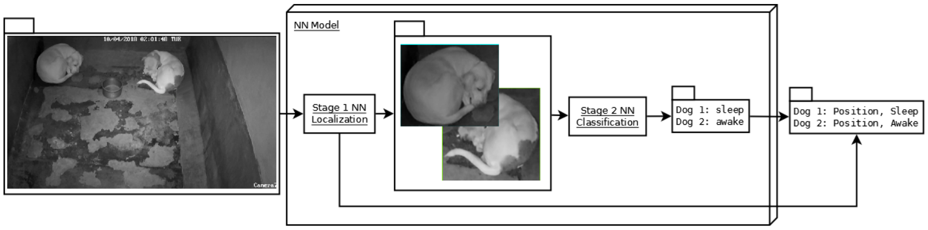

Fig. 3. Two stage model.

We decided to experiment with both types of models, comparing them using the following two criteria.

1. Intersection over union (IoU) is a standard evaluation metric in object detection. Similarly to the approach taken in [24], we calculated the widely used detection accuracy measure, mean Average Precision (mAP), based on a fixed IoU threshold, i.e. 0.5.
2. Number of unrecoverable network errors, i.e., classification errors which are impossible to recover from using the post-processing module (which will be described below). One particularly problematic error is continuous false classification of a non-moving sleeping dog.

While both approaches had comparable results with respect to the first criterion (around 0.75 mAP@0.5IoU on the evaluation set), the second approach performed much better with respect to the second criterion. Therefore, we decided to use the end-to-end architecture[9].

For object detection we used the TensorFlow Object Detection API [25].

Due to a low level of variety in training data we have chosen to use transfer learning based on state-free neural networks pretrained on the COCO dataset

[9] It should be noted that the chosen end-to-end architecture has a drawback of simultaneous detection of the same dog as sleeping and awake due to its detection of two objects (sleeping and awake dog) independently. However, this happens in very rare cases and can be overcome by using a higher confidence level for classification.

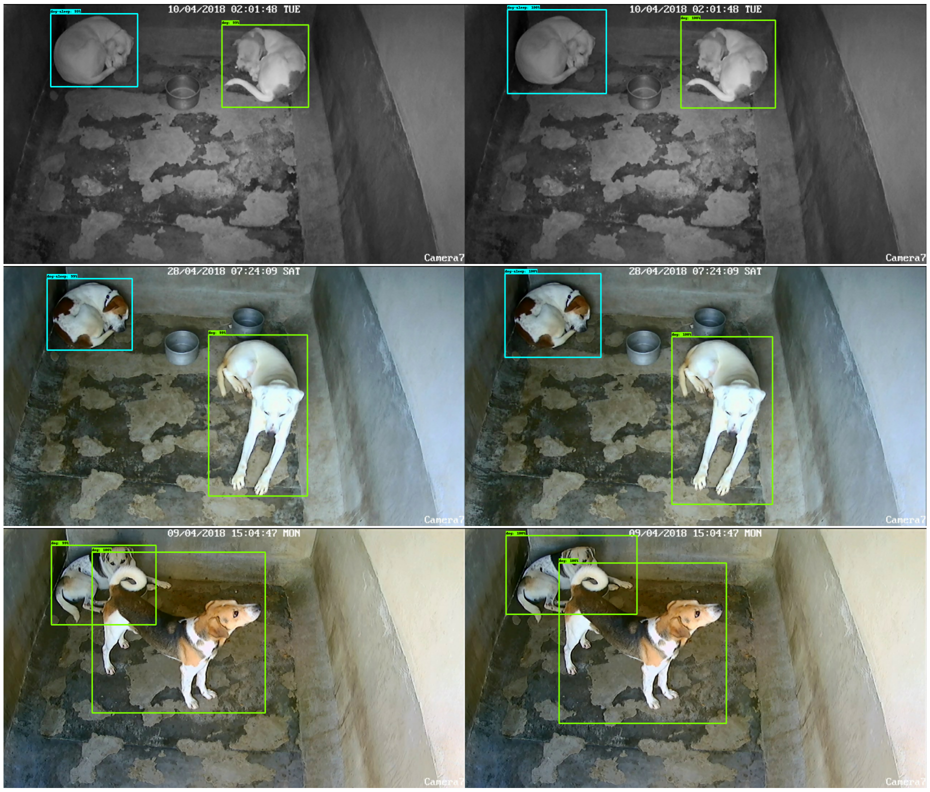

Fig. 4. Example of predicted boxes.

[26]. Initially for better performance we tried to use ssd_mobilenet_v1 [27], but it could not provide sufficient accuracy due to a number of factors, such as a small input dimension. Due to the above, we currently use faster_rcnn_resnet101 [28].

We show some samples of predicted bounding boxes of dogs in the validation set as Fig. 4 where the left column contains the model's prediction, while the right one is the ground truth as annotated by humans.

The output of the neural network consists of N tuples of the form $< x_1, y_1, x_2, y_2, R_{sleep}, R_{awake} >$, where x_1, y_1, x_2, y_2 are the bounding box coordinates, and R_{sleep}, R_{awake} are confidence scores for sleeping and awake dog respectively. This output is then tranformed to $< Ind, x_1, y_1, x_2, y_2, R, Type >$, where $Type$ can be "sleep" or "awake" and $R = \max(R_{sleep}, R_{awake})$, Ind is the dog's index.

4.3 Post-processing

The main idea behind post-processing the network's outputs is compensating for possible errors produced by the network in the tasks of localization and

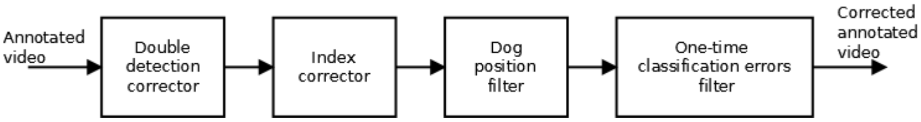
Annotated video
Double detection corrector
Index corrector
Dog position filter
One-time classification errors filter
Corrected annotated video

Fig. 5. Post-processing filter sequence.

classification. The possible errors include: double detection, random detection order, high frequency noise in bounding box coordinates, one-time classification errors, and false-positive sleep detection (in some cases).

The post-processing module consists of a sequence of filters handling a variety of tasks, related to the above mentioned errors. The order of the filters is important due to their non-linearity. For example, it is important to eliminate double detection first, as it may result in a wrong number of detected dogs, which affects further tasks. Fig. 5 presents an overview of the data flow in the post-processor module.

The input of the post-processor is a sequence of images paired with the annotations predicted by the neural network, where:

$$A = < \overline{A^{dog}}, Image > \tag{1}$$

$$\overline{A^{dog}} =< Ind, x_1, y_1, x_2, y_2, Rate, State > \tag{2}$$

and $Image$ is a $1280x720x3$ matrix. A_i denotes an annotated pair for frame i.

The correction tasks performed by the post-processing module as the following (in this order):

1. *Double detection correction* - based on the assumption that the euclidean distance $D(C_i, C_j)$ as in equation (3) between the centers C of detected boxes calculated as per (4) in the instance of double detection (between box i and j) is smaller than some ϵ, and that the probability of this situation for different dogs is quite small. The ϵ parameter is tunable.

$$D(C_i, C_j) = \sqrt{(x_i^c - x_j^c)^2 + (y_i^c - y_j^c)^2} \tag{3}$$

$$C = (x^c, y^c), x^c = \frac{x_1 + x_2}{2}, y^c = \frac{y_1 + y_2}{2} \tag{4}$$

   We calculate D on all pairs of detected boxes for the current frame, and if $D(i, j) < \epsilon$ we compare the detection rate R_i and R_j of these two boxes and delete the one with the smaller rate.

2. *Index correction* - intended for correcting random order of dog indexes Ind in the frame annotation. The index corrector works in the time domain.
   The first step of index correction is calculating the centers of bounding boxes, this data is provided by the previous step.
   The second step is calculating distance as equation (3) between boxes on step k and $k - 1$. At this moment we have a square matrix of distances:

$$\overline{D} = \begin{bmatrix} D(C_1^k, C_1^{k-1}) \; ... \; D(C_n^k, C_1^{k-1}) \\ D(C_1^k, C_2^{k-1}) \; ... \; D(C_n^k, C_2^{k-1}) \\ ... \quad ... \quad ... \\ D(C_1^k, C_n^{k-1}) \; ... \; D(C_n^k, C_n^{k-1}) \end{bmatrix} \tag{5}$$

For each column of this matrix we look for the minimal element and obtain a row of new indices $\overline{Ind_{new}}$. Then the the Ind values in frame annotations are overwritten with new values from $\overline{Ind_{new}}$.

3. *Dog position filtering.* Video can contain different high frequency noises, but the typical neural network is not totally noise invariant, therefore we should use a low frequency filter for position outputs to compensate for the shaking effect of bounding boxes on the output video.

   We use a moving average (MA) filter which s widely used as an indicator in technical analysis that helps smooth out values by filtering out the noise from random fluctuations. It is a trend-following, or lagging, indicator because it is based on past values. In our case we use the following difference equation:

$$P[k] = \frac{1}{n}\sum_{i=0}^{n-1} P[k-i] \tag{6}$$

   where $P = [x_1, x_2, y_1, y_2]$ are bounding box coordinates, n is the filter order (we are using $n = 5$) and all operations are element-wise.

   For an example of filtering, we can look at a plot of the x_1 coordinate for a sleeping dog in Fig. 6, where the orange line represents the non-filtered value, and the blue line is the value after filtering. The same effect applies to the values of the remaining coordinates. This transformation eliminates the jittering effect, providing the user with a more comfortable watching experience.

4. *One-time classification errors filtering.* One of the fundamental features of deep neural networks is the presence of singular points where the output value may be incorrect. Often these points can be artificially obtained by adding manually crafted noise-like signals.

   To compensate for this effect, we use two approaches. The first one is related to motion analysis. We use motion analysis techniques based on classical computer vision methods like Gaussian blur, frame delta calculation, finding contours, etc.

   For detecting movement we use a threshold based method containing the following steps: (a) crop current and previous image to dog bounding box (with coordinates from last frame); (b) convert to gray-scale; (c) calculate absolute difference between cropped images; (d) binarize image by threshold; (e) apply dilation procedure to the image for filling holes; (f) find contours on dilated image.

   If a sufficiently large contour was found, we interpret that as evidence that the dog moved and change the dog's state to awake. This helps fixing false positives in sleeping state classification.

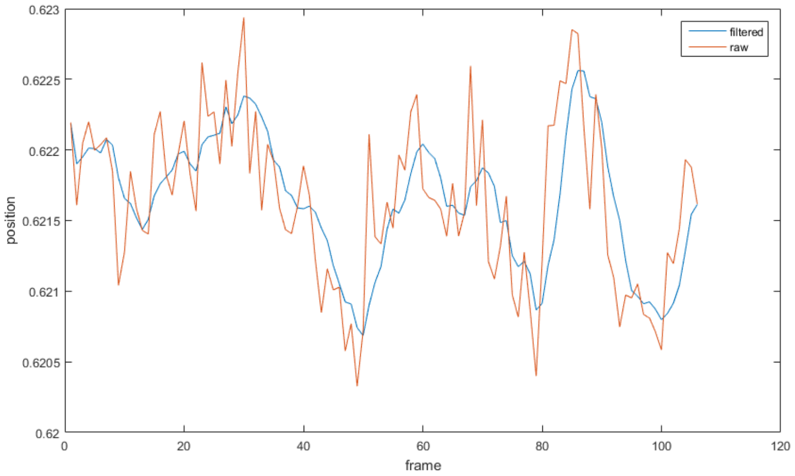

Fig. 6. Example of filter effect.

The second approach is filtering states. This algorithm aims to correct sequences of one state type (i.e. only sleep or only awake), that are shorter than a certain threshold. In most situations this kind of wrongly classified sequence is shorter than 3 frames. This is based on the assumption that the frequency of alternating between asleep/awake states in animals cannot be too high.

We use an approach based on remembering the currently active state of a frame sequence and switching to a new state only after seeing N frames with that state. At first glance, it may seem that this can corrupt the statistics about dog sleeping patterns, but the algorithm is symmetric in regards to states, thus the loss of the previous state's points in the beginning of a new state sequence (we had to wait N frames until toggling the state) is compensated by additional state points after the end of the sequence.

4.4 Sleep Parameters Calculation

Next we describe the calculation of the following parameters: (i) total amount of sleep, (ii) sleep interval count, and (iii) sleep interval length.

We represent a vector of dog states $State$ for dog j as follows:

$$State^j[k] = \begin{cases} 1 \ \text{ if } A_k[A_j^{dog}][State] = "sleep" \\ 0 \text{ if } A_k[A_j^{dog}][State] = "awake" \end{cases} \quad (7)$$

where k is the frame index.

The total amount of sleep for dog j is obtained as follows:

$$Length_j = \sum_{k=0}^{len(State^j)} State^j[k] \tag{8}$$

Sleep interval count is calculated as follows:

$$Count_j = \sum_{k=0}^{len(State^j)} \max(\Delta State^j[k], 0) \tag{9}$$

where $\Delta State^j[k]$ is defined as:

$$\Delta State^j[k] = \begin{cases} State^j[k] & \text{if } k = 0 \\ State^j[k] - State^j[k-1] & \text{if } k > 0 \end{cases} \tag{10}$$

5 Evaluation

We evaluated the system on 10 videos of total length 600 sec. The video set included videos with 0-2 dogs, day/night time and different dogs and rooms. The videos were processed by the system and a testing set of 6,000 frames annotated with the system's predictions were manually checked for correctness by the authors; in controversial cases consensus was reached by discussion between the authors. The manual revision process yielded a result of 5,340 correct frame classifications.

6 Summary and Future Work

Despite dogs being a well studied species in animal science, very few works addressed so far the challenge of automatic analysis of dog behavior. In this paper we presented a system for automatic quantification of sleeping patterns of kennelled dogs, which is being currently used to measure welfare indicators in an ongoing research project. Due to the immense amount of video footage collected in the project, manual analysis is an extremely time consuming, tedious and error-prone task, to which our system, based on convolutional neural networks, provides an efficient and accurate solution. The approach presented here is based on frame vt frame analysis. One of the future research directions is to investigate more sophisticated approaches in which dependencies over time can be modelled (e.g., recurrent systems or modelling dog sleeping-states and frame dependencies using probabilistic models).

More generally speaking, behaviour analysis plays a major role in animal welfare science [29]. Our system demonstrates the potential of using neural networks for revolutionizing the way animal scientists work today. The development of automatic systems for behavior analysis has the potential for impacting the welfare of companion, farm and zoo animals, which is a problem of increasing interest for the modern society. Therefore, an important direction for future research is making the suggested approach generalizable to other types of behavior analysis and other types of animals. Some first steps were already taken in [14].

7 Acknowledgement

This work has been supported by the NVIDIA GPU grant program.

References

1. D. Arney, "What is animal welfare and how is it assessed?," *Sustainable Agriculture*, p. 311, 2012.
2. T. Burghardt and J. Ćalić, "Analysing animal behaviour in wildlife videos using face detection and tracking," *IEE Proceedings- Vision, Image and Signal Processing*, vol. 153, no. 3, pp. 305–312, 2006.
3. P. Ahrendt, T. Gregersen, and H. Karstoft, "Development of a real-time computer vision system for tracking loose-housed pigs," *Computers and Electronics in Agriculture*, vol. 76, no. 2, pp. 169–174, 2011.
4. R. Tillett, C. Onyango, and J. Marchant, "Using model-based image processing to track animal movements," *Computers and electronics in agriculture*, vol. 17, no. 2, pp. 249–261, 1997.
5. D. Sergeant, R. Boyle, and M. Forbes, "Computer visual tracking of poultry," *Computers and Electronics in Agriculture*, vol. 21, no. 1, pp. 1–18, 1998.
6. L. P. Noldus, A. J. Spink, and R. A. Tegelenbosch, "Computerised video tracking, movement analysis and behaviour recognition in insects," *Computers and Electronics in agriculture*, vol. 35, no. 2, pp. 201–227, 2002.
7. H. Van de Weerd, R. Bulthuis, A. Bergman, F. Schlingmann, J. Tolboom, P. Van Loo, R. Remie, V. Baumans, and L. Van Zutphen, "Validation of a new system for the automatic registration of behaviour in mice and rats," *Behavioural processes*, vol. 53, no. 1, pp. 11–20, 2001.
8. A. Spink, R. Tegelenbosch, M. Buma, and L. Noldus, "The ethovision video tracking systema tool for behavioral phenotyping of transgenic mice," *Physiology & behavior*, vol. 73, no. 5, pp. 731–744, 2001.
9. J. J. Valletta, C. Torney, M. Kings, A. Thornton, and J. Madden, "Applications of machine learning in animal behaviour studies," *Animal Behaviour*, vol. 124, pp. 203–220, 2017.
10. C. Palestrini, M. Minero, S. Cannas, E. Rossi, and D. Frank, "Video analysis of dogs with separation-related behaviors," *Applied Animal Behaviour Science*, vol. 124, no. 1, pp. 61–67, 2010.
11. S. Barnard, S. Calderara, S. Pistocchi, R. Cucchiara, M. Podaliri-Vulpiani, S. Messori, and N. Ferri, "Quick, accurate, smart: 3d computer vision technology helps assessing confined animals behaviour," *PloS one*, vol. 11, no. 7, p. e0158748, 2016.
12. P. Pons, J. Jaen, and A. Catala, "Assessing machine learning classifiers for the detection of animals behavior using depth-based tracking," *Expert Systems with Applications*, vol. 86, pp. 235–246, 2017.
13. S. Mealin, I. X. Domínguez, and D. L. Roberts, "Semi-supervised classification of static canine postures using the microsoft kinect," in *Proceedings of the Third International Conference on Animal-Computer Interaction*, p. 16, ACM, 2016.

14. D. Kaplun, A. Sinitca, A. Zamansky, S. Bleuer-Elsner, M. Plazner, A. Fux, and D. van der Linden, “Animal health informatics: Towards a generic framework for automatic behavior analysis,” in *Proceedings of the 12th International Conference on Health Informatics (HEALTHINF’19)*, 2019.
15. S. Amir, A. Zamansky, and D. van der Linden, “K9-blyzer-towards video-based automatic analysis of canine behavior,” in *Proceedings of Animal-Computer Interaction 2017*, 2017.
16. J. Alcaidinho, G. Valentin, N. Yoder, S. Tai, P. Mundell, and M. Jackson, “Assessment of working dog suitability from quantimetric data,” in *NordiCHI’14, Oct 26Oct 30, 2014, Helsinki, Finland.*, Georgia Institute of Technology, 2014.
17. J. Alcaidinho, G. Valentin, S. Tai, B. Nguyen, K. Sanders, M. Jackson, E. Gilbert, and T. Starner, “Leveraging mobile technology to increase the permanent adoption of shelter dogs,” in *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 463–469, ACM, 2015.
18. A. Zamansky, D. van der Linden, I. Hadar, and S. Bleuer-Elsner, “Log my dog: perceived impact of dog activity tracking,” *IEEE Computer*, 2018.
19. A. Zamansky and D. van der Linden, “Activity trackers for raising guide dogs: Challenges and opportunities,” *IEEE Technology and Society*, vol. 37(4), pp. 62–69, 2018.
20. D. van der Linden, A. Zamansky, I. Hadar, B. Craggs, and A. Rashid, “Buddy’s wearable is not your buddy: Privacy implications of pet wearables,” *forthcoming in IEEE Security and Privacy.*
21. C. Ladha, N. Hammerla, E. Hughes, P. Olivier, and T. Ploetz, “Dog’s life: wearable activity recognition for dogs,” in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 415–418, ACM, 2013.
22. R. Brugarolas, R. T. Loftin, P. Yang, D. L. Roberts, B. Sherman, and A. Bozkurt, “Behavior recognition based on machine learning algorithms for a wireless canine machine interface,” in *Body Sensor Networks (BSN), 2013 IEEE International Conference on*, pp. 1–5, IEEE, 2013.
23. L. Gerencsér, G. Vásárhelyi, M. Nagy, T. Vicsek, and A. Miklósi, “Identification of behaviour in freely moving dogs (canis familiaris) using inertial sensors,” *PloS one*, vol. 8, no. 10, p. e77814, 2013.
24. M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.
25. J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” *CoRR*, vol. abs/1611.10012, 2016.
26. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014.
27. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.
28. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.
29. M. Dawkins, “Using behaviour to assess animal welfare,” *Animal welfare*, vol. 13, no. 1, pp. 3–7, 2004.