Monitoring Dog Movement When Left Alone

Daniel Raab, Jacob Ueltzen, Konrad Creutzburg
Faculty of Engineering
Leipzig University of Applied Sciences
Leipzig, Germany

***Abstract*—A monitoring system for dogs being left alone at home is developed using a Raspberry Pi, a webcam and computer vision algorithms. The system can monitor a room with a dog and evaluates whether the dog is in a calm or stressed state. It sends a notification to the owner when stress is detected.**

***Index Terms*—Dog monitoring, dog left alone, motion detection, background subtraction**

I. Motivation

Most dog and puppy owners leave their dogs alone at home regularly to go to work, university or to get groceries. To be able to do so, dogs and puppies must first be trained to be left alone. Otherwise, the dog is likely to experience distress or even develop separation anxiety. Distressed dogs can manifest many symptoms e.g. barking, whining, pacing, circling, damaging furniture or try to escape [1]. Therefore, training dogs, and especially puppies, to be comfortable when being left alone is important. It is usually done with the following strategy: the owner starts leaving the dog alone for short intervals of time, a minute for example. Then, step by step the time is increased when the dog can handle the time interval without distress. It is important that the dog is trained to be calm and at rest when being left alone. This means that distress and activity during training must be avoided, because this is unwanted behaviour. In general, dogs should be resting and lying down most of the time. T. Rehn and L. J. Keeling [2] analysed 90 hours of video material of dogs with no separation related behaviour problems. They found, that twelve different dogs spent 95% of the time lying down.

This shows, that calm dogs spend most of their time lying down when being left at home. To aid in the training and monitoring of dogs being left alone, a monitoring system is developed. Its purpose is to recognise and report dog unrest and activity. The system should use a cheap single board computer and a webcam to monitor the room with the dog. Furthermore, it should detect motion via computer vision algorithms. This indicates, that the dog is in a distressed condition. The owner can be notified on his smartphone, to intervene and help the dog if necessary. This allows easier training of puppies and dogs by giving the owners more feedback of whether the dog is calm or in distress when being left alone.

II. Method

A prototype system is developed using a Raspberry Pi and a Logitech C200 webcam. The test subject is a puppy at the age of 18 months. The positioning and orientation of the camera is adjusted to include most areas in the frame where the dog can linger. To detect movement of the dog in the frames recorded with the camera, multiple assumptions are made:

- static viewpoint from a stationary camera
- all movements in the room are movements of the dog

A. Movement detection

With these assumptions, detecting dog movement can be implemented by looking for changes between the current frame and the previous frames. A suitable method for this implementation is the **Background Subtraction**. The static background is subtracted from the image, extracting the foreground. Another possible implementation would be the use of neural networks to detect the dog as an object in the frame. However, developing a system of this kind is more complex, needs significantly more hardware resources and might not be as robust when the dog is partially obstructed.

Figure 2 demonstrates the operating principle of the background subtraction on the given example. The first image shows a frame of the room without the dog. The second image shows a frame where the dog has just moved to a different spot. The output of the background subtraction method from OpenCV [3] [4] is shown in the third image. The dog shape is clearly visible except for the partially obstructed parts.

This background subtraction method does not use a static background image, but adapts it dynamically. This allows for changes in the background, such as variation of the ambient lighting. It also allows the dog to become part of the background after lying still for a specific period of time. How fast the background adapts, and therefore how fast a non-moving object is integrated into the background, can be configured.

The background subtraction creates a binary image where the detected foreground is shown in white and the background in black. This so called mask shows detected movement in our use case. Because of noise in the camera and lighting changes, there is noise in the background subtraction image, which can be seen in the third image of Figure 2. To remove the noise, the morphological operation of opening is used. Opening is erosion followed by dilation and removes noise. A kernel of 4x4 was empirically established. The result can be seen in the last image of Figure 2.

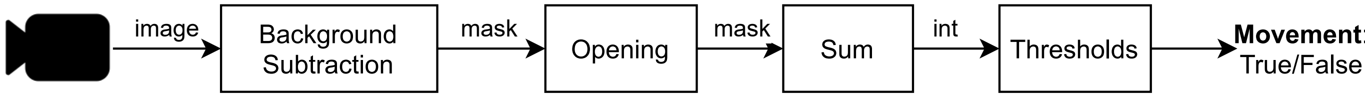

Fig. 1. Processing steps of the developed algorithm

Fig. 2. Images from the algorithm processing steps (from top to bottom): webcam image of the room, webcam image with dog, background subtraction output, filtered background subtraction output

B. Assessment of foreground activity

The next step is to classify each frame with ”rest” or ”unrest”. After applying the opening filter, there is a binary image with white areas indicating movement in the foreground. Hence, the amount of white in the image is a measure of movement. By summing all pixel intensities of the image, an integer value is obtained. This is shown in the ”Sum”-step in Figure 1. Because of the remaining noise the intensity sum will not always be zero when the dog is resting. To counteract this interference a threshold was empirically determined. Images with an intensity below this threshold are classified as “rest”. To filter out interference caused by lighting change, a second cutoff threshold is introduced. When the lighting changes, the majority of the image will show a change and thus the sum intensity is a lot higher than the movement of a dog. This is shown in Figure 3, where the intensities of all frames are plotted as a histogram. The dotted vertical lines show the thresholds and the colors show the categorisation of the intensities: blue is rest, orange is movement and green is a lighting change, which is also classified as rest.

III. Evaluation of the algorithm

A. Video Material

The algorithm was developed by analysing and testing with video material recorded with a webcam. To have enough data — especially dog movement data — multiple videos were recorded (and sometimes split into smaller videos). The result is ten videos with a combined length of about 3.5 hours.

B. Annotation

The video material was annotated with the open source video annotation tool *VGG Image Annotator (VIA)* [5]. When dog movement was apparent in the video, annotations were added. The classification of whether or not the dog is moving is a matter of subjective judgement. Especially the beginning and end of a movement event are difficult to annotate. This affects the accuracy of the metric as discussed in section V.

C. Metric

From the manually performed annotations, every frame was classified into “rest” or “unrest”. This is the reference data used to evaluate the algorithm performance. It can be compared to the output of our algorithm. This comparison of reference data and algorithm output is performed frame by frame. It provides a measure for performance of the developed algorithm for each input video. The performance is measured in true positives and false positives.

Video	frames (total)	movement	true positives	false positives	true positive rate
1	8396	1440	1278	684	88.8 %
2	11187	254	161	309	63.4 %
3	12767	474	279	766	58.9 %
4	22522	695	644	1105	92.7 %
5	7029	200	120	209	60.0 %
6	7097	288	247	388	85.8 %
8	7956	86	58	88	67.4 %
9	4572	559	352	396	63.0 %
10	4669	443	259	156	58.5 %
Sum	**86195**	**4439**	**3398**	**4101**	**76.5 %**
7	6761	321	1	568	0.3 %

TABLE I
ANNOTATED DATA AND ALGORITHM DATA

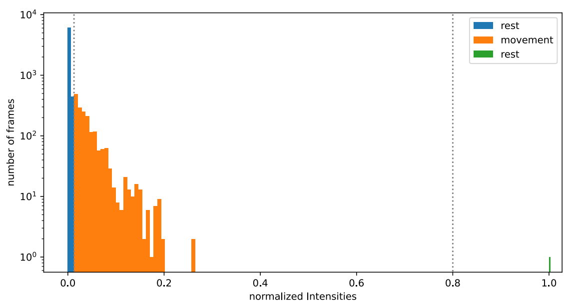

Fig. 3. Histogram of the intensities of all the frames. Split according to the thresholds (dashed vertical lines) into rest, movement and rest again.

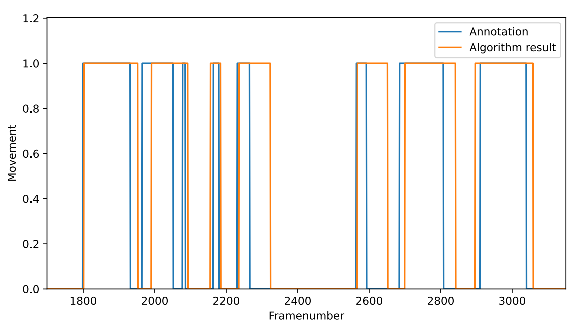

Fig. 4. Cutout from the plotted annotated frames and the frames marked by the algorithm (1 is movement, 0 is rest)

IV. RESULTS

A computer vision algorithm was implemented to classify frames from a webcam video stream into two categories: dog **rest** and dog **unrest**.

Table I shows the results of ten videos. One outlier (Video 7) was removed for further evaluations, because the annotated data is faulty (the annotations were shifted in time). As stated in section I the main point of interest are the true positives. The algorithm detected a total of 3398 frames with movement of the 4439 annotated frames. Therefore the true positive rate is 76.5 %. 4101 frames were detected as false positive, which is 1.2 times as many false positives as true positives.

At first glance, these results do not seem great, but they have to be put into context. We are comparing frame by frame, but, as stated in subsection III-B, the annotations of movement are subjective and difficult to determine.

Putting the results into context, it is important to point out the inaccuracy of the reference data. As elaborated in subsection III-B especially the beginning and end of a movement cannot always be set objectively. The results therefore not only give a measure of the algorithm performance, but also include the flaws of the reference data. A true positive rate of 100 % with no false positives or false negatives is unobtainable.

In order to illustrate this effect, a fraction of a video was analyzed in Figure 4. The figure shows a time plot with the annotation and the output of the algorithm. It can be seen that the annotations and the algorithm output mostly overlap, but often differ in when the movement starts or ends. Therefore the score for this fraction of the video is not a perfect score. However, the small differences are not relevant to the purpose our use case.

V. DISCUSSION

When reviewing the video and algorithm output side by side, and considering the true positive rate of 76.5 %, the result is sufficient for the use case of detecting dog distress and movement. We determined empirically, that the threshold is best to be placed relatively low, making the algorithm rather "trigger happy". This typically results in a higher false positive rate. Considering the benefits of a higher true positive rate, this is a good trade-off for the use case.

The following **limitations** apply for the introduced system. The setup is specific to our testing environment. This includes the positioning of the camera in the room as well as the region of interest in the frame. Thresholds were hard coded to match the ambient light of our testing environment. Changes of the ambient light due to weather are present in the given data set. However, a low light case (night time) was not tested. Finally, data from only one puppy was tested. Impact of the behaviour of the test subject cannot be evaluated.

A. *Real-Time Suitability*

The algorithm was tested with a Raspberry Pi 4 and a Logitech C200 webcam with 640x480 resolution and a framerate of 7.5 images per second. A telegram bot was added to send out notifications when movement is detected. Because the algorithm is not computationally intensive, this system easily handles the 7.5 frames per second of the webcam. An example notification of a detected movement is shown in Figure 5. At the beginning and the end of a detected movement the current image of the webcam is sent to the user.

Fig. 5. Telegram bot report on movement

VI. Conclusion

A application for a dog-movement detection was developed. It is based on the background subtraction method to minimize computational requirements compared to an AI-based approach. The presented algorithm runs on a single-board computer processing a live video feed from a connected webcam. Therefore it is well suited for the task.

References

[1] B. L. Sherman, “Separation anxiety in dogs,” *Compendium*, vol. 30, no. 1, pp. 27–42, 2008. [Online]. Available: https://www.canisbonus.com/wp-content/uploads/2012/07/Sherman-Understanding-separation-anxiety-2008.pdf

[2] T. Rehn and L. J. Keeling, “The effect of time left alone at home on dog welfare,” *Applied Animal Behaviour Science*, vol. 129, no. 2-4, pp. 129–135, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168159110003242

[3] “Opencv: How to use background subtraction methods,” version 4.5.4. [Online]. Available: https://docs.opencv.org/4.5.4/d1/dc5/tutorial_background_subtraction.html

[4] Z. Zivkovic, “Improved adaptive gaussian mixture model for background subtraction,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 2004, pp. 28–31 Vol.2.

[5] A. Dutta and A. Zisserman, “The VIA annotation software for images, audio and video,” in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM ’19. New York, NY, USA: ACM, 2019. [Online]. Available: https://doi.org/10.1145/3343031.3350535