Improving the Annotation Efficiency for Animal Activity Recognition using Active Learning

S.J. Spink*[1], J.W. Kamminga[1] and A. Kamilaris[1,2]

1. Department of Pervasive Systems, University of Twente, Enschede, the Netherlands

2. CYENS Center of Excellence, Nicosia, Cyprus

*** Corresponding author: suzanne.spinik@gmail.com**

Abstract

Animal activity recognition (AAR) is essential for the conservation of endangered species and the well-being of livestock. Small resource-constraint devices using inertial measurement units can be attached to animals to automatically classify the performed activities such as running or eating based on their bodily motions. The IMU time-series data is enriched with ground-truth annotations by experts to train artificial intelligence models as classifiers. Annotating AAR data is tedious and time-consuming. Experts that can provide high-quality annotations are scarce and their time is costly. It is difficult for annotators to determine what data is essential to annotate, thus they tend to annotate large amounts of data to improve the performance of the trained classifiers. In this paper, we show that active learning (AL) increases annotation efficiency by selecting only the most informative data for annotation and algorithm training. We use real-world IMU data from four horses performing various activities. We compare five AL-based algorithms (three uncertainty sampling and two disagreement-based sampling (DBS)) with random sampling. Our results show that DBS increases the annotation efficiency, especially when the AAR problem is harder by increasing the number of classified activities from 6 to 8. Furthermore, we show that random sampling can be an effective method to improve annotation efficiency when the AAR task is not too difficult.

Introduction

Animal activity recognition (AAR) is essential for the conservation of endangered species and the well-being of livestock. The activity of animals is a rich source of information that not only provides insights into their life and well-being but also their environment. Due to the advent of small, lightweight, and low-power electronics, we can attach unobtrusive resource-constrained devices to animals that measure a wide range of aspects such as location, temperature, and activity. These aspects can be used to support numerous application domains, including wildlife monitoring, anti-poaching, and livestock management.

Many machine learning (ML) models that are used for AAR are supervised and require annotated datasets for training and evaluation purposes [1], [2]. AAR is particularly a use case where active learning (AL) may have a huge impact. AAR is harder than human activity recognition (HAR) because it is difficult to observe animal behavior in the wild, while their activity patterns are not always known beforehand. Because we cannot ask (wild) animals to perform all the activities that should automatically be recognized, the resulting dataset has to be huge. Wild animals have to be monitored for long periods to opportunistically record the activity of interest. For example, African wild dogs may only eat three times a day, while they can devour an antelope in just 15 minutes. As a result, AAR datasets can be very large and severely imbalanced. Therefore, it is tedious for annotators to find those segments of data that should be annotated for the best performance of the AAR classifier. Furthermore, the sensor orientations are not fixed and the recorded data is very noisy which increases the difficulty of determining what data is most useful for training [3]. It is difficult for annotators to determine what data is essential to annotate, thus they tend to sequentially annotate large amounts of data to improve the performance of the trained classifiers. The annotation process is tedious, labour intensive, and expensive because the availability of experts is limited and their time costly.

In this paper, we show that AL increases annotation efficiency by selecting only the most informative data for annotation and algorithm training. We consider a pool-based AL [4]– [7] setting. In pool-based AL there exists a small set of labeled data and a large pool of unlabeled data. The idea behind AL is to select only those samples - to be annotated - from the pool so that the performance of the classification task is maximized. This smarter way of querying optimises the training process by helping the ML model to learn and converge faster. In practise, a query occurs by asking a human ’oracle’ for the correct label and adding the annotated sample to the training set. As a result, classification can become significantly faster and cheaper.

We use real-world inertial measurement unit (IMU) data recorded using four horses performing various activities during a week at an equestrian facility. This paper focuses on the impact AL has in the field of AAR [6], [8]–[10], considering the case of horses and activity data recorded by a three-axis accelerometer placed on the horses. To the knowledge of the authors, this is the first effort to investigate AL in AAR using IMU sensors/data.

Related Work

This section includes all essential empirical research on activity recognition and AL, including the different types, strategies, and techniques that have been used.

Active Learning Algorithms

One way of deciding which instance to label next in a data set, is by random sampling. However, there are limited experts in this area that can annotate this data well, which makes the process costly. This makes it more important to not waste their time by randomly annotating unnecessary instances. With AL, the most ambiguous instance is found to be annotated, to maximise the efficiency. This instance is added to a training set and classified again. This process is done iteratively, each time asking the annotators to label the most ambiguous instance. There are two main divisions in AL algorithms mostly used, namely *Uncertainty sampling* and *Disagreement-based sampling*. Both are described below.

Uncertainty Sampling: Uncertainty sampling (UNCS) gives an intuitive view into how AL works [4]–[6] and it has a low computational complexity [7]. It needs to be combined with a classifier. Many successful applications can be found in natural language processing (NLP) tasks, which require enormous amounts of data and labelling costs are consequently high. Dredze and Crammer [11] used the confidence margin technique of UNCS on four different NLP tasks and compared the results with random sampling and margin sampling. Accuracy of AL was significantly higher than random sampling and needed only 73% of the labels that random sampling needed. Nonetheless, Zhu [12] found that UNCS does not work well if there are many or significantly large outliers, which are not useful for the model to learn from, making it harder for the model to converge.

Disagreement-Based Sampling: Within disagreement based sampling (DBS), the *committee of classifiers* algorithm is used most often. Muslea [10] maximized the efficiency of DBS in finding the correct label, by introducing co-testing. This is a combination of a committee of classifiers, training all the classifiers at the same time. Several classifiers are used and then the instances where they disagree on the most, called *contention points*, are harnessed to train the classifier. However, co-testing can also be unfavourable [13], in case future data acquisition gives substantially different data instances, e.g. new activities are introduced.

Activity Recognition

Many methods and algorithms have been applied to optimize classification in AAR, however, very limited research has been conducted on AL applied in AAR. However, AL has been applied more widely in Human Activity Recognition (HAR) in the past. This is a similar field, but the problem is easier since annotating humans and their activities is more straightforward, while it is easier to recruit annotators. Animals cannot give feedback, move unexpectedly and especially the behavioural pattern of wild animals in their natural habitat is still widely unknown. This makes classification more difficult.

Animal Activity Recognition: Many classifiers have been proposed for AAR. Support Vector Machines (SVM) has been used by Sturm et al. [2] for a similar purpose, classifying six activities based on IMU data. Furthermore, it Gao et al. [1] used not only 3D accelerometer data but also videos. Both spatial features (i.e. standard deviation and signal magnitude area) and frequency-domain features were extracted.

Human Activity Recognition: AL has been applied widely in HAR. Pool-based sampling was used for almost all experiments while the algorithm type varied between them. Stikic [8] used a combination of pool-based sampling with two different algorithms, UNCS and DBS, to classify ten activities. For UNCS, two samples were chosen each time iteratively, while for DBS, one sample with the highest disagreement was chosen. The results showed little difference in accuracy between the two, but both saw a large increase in accuracy when the number of labelled instances increased. Furthermore, Vaith et al. [6] investigated AL with IMU data of humans, by doing a human gait analysis. They observed that the Variation Ratio (VR) and Maximum entropy

(EM) strategies were the most accurate, within the UNCS spectrum. Finally, Liu [9] compared different AL algorithms, concluding that AL with UNCS and DBS perform better than supervised learning and random selection.

Methodology

Activity data was collected by attaching sensors on horses that recorded 3D accelerometer data, as described in [14]. Two AL algorithm types were then applied to the horses' dataset created: disagreement-based sampling (DBS) and uncertainty sampling (UNCS) [4]. Within UNCS, three algorithms were considered: least confident, margin uncertainty and entropy. DBS considered two types: maximum disagreement and consensus entropy. These algorithms will be compared to: each other, the classifier without AL and state-of-the-art.

Dataset and Pre-processing

We use real-world IMU time-series data from the Horsing Around dataset [14]. The data was collected at an equine facility over a time span of seven days. The horses were either located in their stables, in nearby riding areas, or in an outdoor field. On most days, the horses were also allowed a break in another outdoor field where they demonstrated more natural behaviour such as rolling and grazing. The IMUs sensors used were human activity monitors from Gulf Coast Data Concepts [15]. These sensors include an accelerometer, magnetometer, and gyroscope. Sampling rate was set to 100Hz. Sensors were attached to the horses by means of a collar, located around the middle of the horses' neck. Data was annotated chronologically using a labeling tool[1] [3], which is publicly available online [16]. When a horse was performing multiple activities simultaneously, the activity that was mainly exercised was chosen as the label. The Horsing Around dataset comprises data from 18 horses in total. Because the data was annotated sparsely for most horses, we could only use data from four horses that had sufficient annotations for the same 8 activities. The selected horse names are Galoway, Patron, Happy and Driekus [14].

During the experiments, we used two scenarios, 8 activities and 6 activities to investigate the effect of label granularity on AL, see Figure 1. We used the following 8 activity classes: standing, grazing, head-shake, walking (rider and natural), trotting, and running (rider and natural). In the 6 activity scenario, we dropped head-shake and merged running-rider with running-natural into one running class.

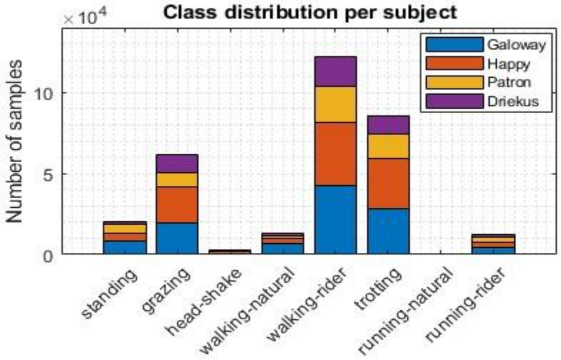

Figure 1: The amount of available data per activity class for each subject. The running-natural class is very small and was merged with running-rider when 6 activities were used for the experiments.

The 3D vector (l2-norm) of the accelerometer data was used to obtain a more orientation-independent input feature. A lowpass Butterworth filter was applied before splitting the data into a pool and test subsets. The features were scaled and data was windowed based on a sliding window of two seconds and 50% overlap, shuffled, reshaped to a one dimensional array and encoded with one-hot encoding in their own subset before classification. This gave a total of 81,332 samples.

Active Learning

We considered five different AL algorithms in total. We compared three types of UNCS algorithms. *Least confident* is used most often in research and it considers a 1-prediction. *Margin of confidence* finds the two highest predictions and subtracts them. This

[1] Note that no AL was used during the annotating of the dataset.

margin then finds the instances that lie closest to it, marking them as the most ambiguous ones. The last was *uncertainty entropy*, which looks for the most random instance. In regards to DBS, two types were chosen. *Consensus entropy* uses the averages of class probabilities per classifier and then finds the entropy. *Maximum disagreement* uses the Kullback-Leibler divergence and finds how both classifiers differ. Since these sampling types have different approaches of finding the next most ambiguous instance, they were all selected and compared for reasons of completeness. As UNCS algorithms use a formula to calculate which instance to query next from the rest dataset, they require only one classifier to train the data (Deep Neural Network, DNN). DBS algorithms require several classifiers, which are used and compared to each other. After a prediction, the classifiers need to decide on the instance with the most disagreement. Hence, these algorithms use two DNN classifiers, both starting with different samples at the initial training set.

Active Learning Variables: Four variables are considered and analysed. The first is the size of the initial training set, which is chosen as 10 instances. The initial training set is selected from the pool subset by randomly selecting instances. This size is very small, so AL can be applied and the effect shown as soon as possible, while still big enough to classify and test. The rest of the data instances from the pool data set will be in an unlabelled subset, known as the *rest subset*. From the initial training data, the pool is used to train the classifier. Then, the most ambiguous instance is selected from the unlabelled rest subset, labelled, and added to the training set. This is performed again and again, based on an iteration number *IT*. The optimal values of these two variables (i.e. *DP* and *IT*) are considered through our experiments. The third variable is the function which finds the most ambiguous instance each time and the fourth variable is the number of activities to classify, which is compared at six and eight activities.

Classification

DNN Classifier: The classifier used is a sequential classifier from the Keras Python DL API, representing a neural network with multiple layers. The first layer is a *Reshape* layer, where training data is reshaped into 6 dimensions. Next, three *Dense* layers are added, representing three hidden layers in the neural network, each with 100 fully connected nodes. The activation function is a rectifier (ReLu activation function). Then, a *Flatten* layer is added to flatten the data and the last layer is the *Output*, which is a dense layer with six or eight nodes (same as the number of horse activities) and a *softmax* activation function.

Evaluation: We used leave-one-subject-out crossvalidation for each algorithm. All AL algorithms were evaluated using four folds, each fold all data from one subject was used as the validation set. Each fold the pool dataset comprised all data from the other three subjects. The respective poolsizes (windows) are 53242 for Galoway, 66647 for Patron, 54435 for Happy and 69735 for Driekus. The F1 score was used to evaluate the performance of the trained classifier in all scenarios. The F1 score is defined as F1=TP / TP+1/2(FP+FN), based on True/False Positives/Negatives, i.e. TP, FP, TN, FP respectively. The testing phase is performed 4× *IT* times, once per tested horse per iteration. To get the results of the F1-score per iteration, the average of the four horses is taken.

Results

The experiments are divided into three sections. First, the algorithms are compared to each other and random sampling based on F1 score and the number of activities classified (Section IV-A). Second, the algorithms are compared to DNN and manual annotation without AL (see Section IV-B). Finally, a comparison with state-of-the-art work is performed (see Section IV-D).

Comparisons of Sampling Types

There is a clear distinction between the two types of AL, see Figure 2, where DBS with the maximum disagreement algorithm and consensus entropy algorithm clearly outperform UNCS with least confident, margin and uncertainty entropy algorithms. When considering the classification of six activities of this dataset, random performs slightly worse than the best AL algorithm (i.e. DBS) based on the F1 score. However, when more sampled labels are used (i.e. from F1 score of 0.5 onwards), this difference in performance decreases.

Effect of number of activities: We investigated the effect of the number of activities, or granularity, on the performance of AL. The AAR task becomes more difficult by having to distinguishes more activities that are very similar to other activities.

Furthermore, in the 8 activity scenario, see Figure 3, the two additional classes were very small minority classes. Random sampling is consistently slightly worse than DBS.

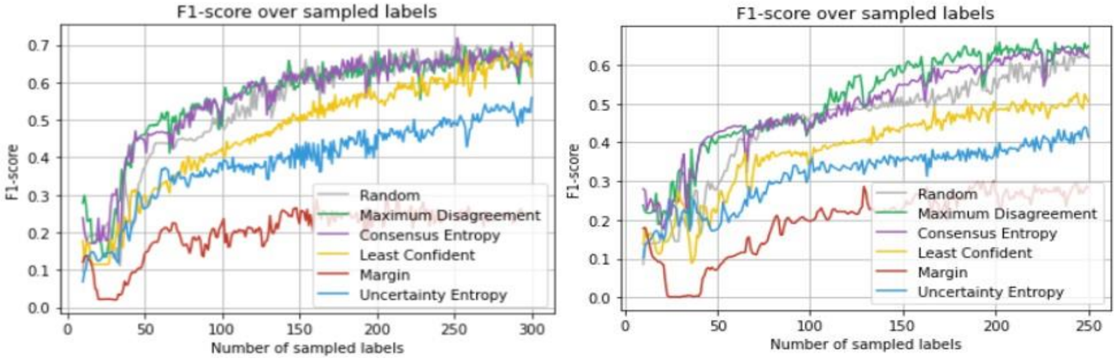

Figure 2 & 3: Comparing least confident, margin, uncertainty entropy, maximum disagreement, consensus entropy and random sampling based on a six-activities classification and eight-activities classification.

For all AL algorithms, it can be observed that the algorithms first learn very quickly and then slow down. For the six activities classification, see Figure 2, a plateau with an F1 score of 0.5 is reached after 50 labelled samples for DBS, while random only reaches this after having used 100 labelled samples. A similar observation holds for the eight-activities classification although, in this case, all AL algorithms and random sampling perform slightly worse in respect to the F1 score, which is a little lower for the same number of labelled samples used. While random sampling and DBS overlap between 70 and 100 sampled labels, DBS performs better for the next additional 150 sampled labels, up and till the F1 score becomes 0.63. Furthermore, when comparing six and eight activity classifications, there can be observed that random sampling is equally good as DBS at a F1 score of approximately 0.6, but it takes more labelled samples to reach this performance when classifying eight activities.

Comparison without AL

The main benefit of AL is shown when considering the number of labelled instances that are used, considering that labelling usually requires significant time. The benefits of using AL instead of manual annotation have already been studied in [17]. It depends on the performance of a human with respect to annotation time, which differs per instance and per data set type.

Via manual annotation, each instance needs to be labelled by hand. This takes 3-10 minutes for sequential annotation tasks (as the one of this paper), with an uncertainty rejection of 5.7% [18]. Research on AAR showed that one minute of video took 3.7 minutes to label on average [18]. The time step of the data instances used in this data set was at 1 second, with a 50% overlap. This gives segments of 2 seconds and the annotation time would therefore lie around 7.4 seconds on average. As observed above, a plateau is reached at a F1 score of 0.5 for six-activities classification, where DBS uses 50 labelled samples and random sampling uses 100 labelled samples. This would suggest saving 50% of labelling time for this data set, which is approximately 6 minutes. However, in practice, a F1 score of 0.5 for AAR is not high enough. Therefore, a comparison of a higher F1 score must be made. For the six-activities classification case, random is equally good as DBS at a higher F1 score. However, for the eight activities classification, this was not the case. The highest F1 score was observed at a F1 score of 0.6. The maximum disagreement algorithm was the first to reach this, using 140 labelled samples. Random sampling reached this after using 240 labeled samples. This saves 42% of labelling time for this particular data set when classifying eight activities, which is translated to around 12 minutes of human effort. What must be noted though is that random seems to overlap again with DBS at a F1 score of 0.6 and a higher F1 score can be reached with more labelled samples, which is often desirable for AAR.

In real-life, multiple annotators are often employed, which requires additional time till they fully familiarize themselves with the problem, which is highly dependent on the data which needs to be annotated. In literature, four methods are identified for finding the reliability of annotators of AR problems [19]. Therefore, the saved annotation time (calculated between 6 and 12 minutes for this problem) is lower than the actual annotation time saved.

Using only supervised learning

Finally, the DNN performance with and without AL was measured, comparing with the best UNCS and DBS algorithms scoring the highest F1 score, namely the least confident and the maximum disagreement in the six-activities classification scenario. This was at a F1 score of 0.7, which maximum disagreement reached by using 253 labelled samples while least confident by using 293 labelled samples. When no AL was used, the DNN classifier used all the labelled data for training. The highest F1 score of 0.723 is achieved when all labeled data instances are used, not using AL. However, the performance is only slightly better than the best performing AL algorithm. The DNN had a F1 score of 0.723, while the least confident algorithm and maximum disagreement 0.703 and 0.697 respectively. The difference is therefore only 0.02 for the least confident and 0.026 for the maximum disagreement.

Comparison to State-of-the-art Work

In literature, most research compared AL only to random sampling [11], [20], [21], not considering AAR. For the higher initial training set sizes, some algorithms performed slightly better than random sampling. This was the case for both DBS and UNCS. For DBS, results were similar to UNCS. This was also the case in this paper. However, UNCS was also compared to supervised learning in literature, where Liu [9] found that AL scored a 4-5% higher classification accuracy (*CA*) than supervised learning. Additionally, Stikic [8] found that co-training with uncertainty sampling had an increase of 0.25 - 0.35 in CA over supervised learning. These findings do not align with what was found in this research, where AL was slightly less efficient than when using the whole labeled set. However, we should note that [8], [9] used CA as a metric and we used the F1-score because it is more appropriate for activity recognition.

Discussion

The best performance was obtained when classifying six activities. Both DBS algorithms substantially outperformed UNCS, especially at the lower number of sampled labels. The best performing DBS algorithm is the maximum disagreement algorithm and the best performing UNCS algorithm is the least confident algorithm. Both reached the same maximum F1 score of 0.7 using 300 sampled labels, but maximum disagreement was much faster in doing so. There can also be observed that random sampling performs well. However, for the lower number of sampled labels, DBS outperforms random sampling and this saves 50% in labelling time at 0.5 F1 score, but from there, random sampling overlaps in performance with DBS. As often a higher F1 score is necessary in AAR, this initial advantage over random sampling may not be significant in practice.

In addition, the DNN without AL still outperforms the DNN with AL, based on a small margin of 0.02 in the F1 score. However, this small margin is not significant, as AL can fluctuate based on many factors, e.g. the data used and the subsequent samples queried. In addition, the DNN needed all 81,332 labeled samples for classification, while AL only needed 293 labeled samples. This saves approximately 166 hours of manual annotation time, although this comparison may not be as fair, considering that not all of the 81K labels would be necessary. The work performed in this paper shows the potential that AL has for applications where some small error can be accepted in the overall accuracy. However, as state-of-the-art work shows, the performance can be improved even more by fine-tuning some variables more properly [17].

Recommendations

The AAR performance can be improved by elaborating the fine-tuning and pre-processing steps. Since outliers were present in the dataset used, affecting overall results, those outliers could be detected and removed, or not taken into account in the AL process. If an outlier is picked during an AL query, the model might learn that a rare deviation is normal and will try to compensate all other classifications. Therefore, avoiding outliers may improve performance. Furthermore, the sensors attached to the horses were not completely fixed. This means that the data collected may have been noisy. While a low-pass Butterworth filter was applied to compensate for this noise, this filter did not remove the noise completely. Finally, it would be beneficial to run the experiments multiple times, considering more activities and similar datasets, to validate and compare our findings.

Future Work

The recommendations mentioned in Section V-A will be addressed in future work, to give a more complete picture of the potential effect of AL in the domain of AAR. Guidelines, lessons learned and best practices should be defined and formalized, to help scientists harness AL in the most profitable way in their projects. These practices include also the preprocessing steps required, such as the best combinations of initial training set sizes and optimal number of iterations, which might need to be linked to AL algorithms and datasets used. Additionally, some other aspects of AL were not applied in this study, but do have the potential for future use. For example, other classifiers beyond the DNN classifier used could be employed, e.g. SVM, random forests, and more advanced DNN architectures. This is especially useful for the case of DBS, where a focus could be on finding the effect of different combinations of classifiers. The performance of this algorithm can in this way be optimised fully. Furthermore, it was established that the number of activities to be classified has a big influence on the effectiveness of AL. To quantify the precise influence, more research needs to be performed examining different numbers of classes and types of instances, e.g. overlapping or clear differences within the classes. Moreover, other types of algorithms within both UNCS and DBS can be considered. To understand the full potential of AL, a more in-depth look must be taken.

Conclusion

This paper employed active learning as a promising technique for reducing the time and effort needed for manually annotating data required for training supervised-based models, in the domain of animal activity recognition based on IMU data. The paper demonstrated the use of AL considering horses and classification of their activities, recorded by means of three-axis accelerometer sensors placed around their necks. A deep neural network (DNN) was used as a classifier, together with AL, in order to classify six and eight activities of the horses. Different AL strategies were considered, including three uncertainty sampling algorithms and two disagreement based sampling algorithms. The overall findings on the horses' dataset indicated that AL would have saved a person a lot of effort and labelling time.

A clear preference for DBS over UNCS is found for this data set, where DBS was consistently higher. Furthermore, classifying fewer activities (six) gave a higher performance than classifying more activities (eight). In addition, AL shows an improvement of 50% in annotation effort at a low F1 score compared to random sampling within this six activity classification. However, at higher F1 scores, which is often desirable with AAR, more labeled samples are necessary and AL is not better than random sampling. The potential of AL is higher when classifying more activities, as DBS consistently performed better than random sampling when classifying eight activities for longer. However, again, at a higher F1 score, random sampling and DBS did overlap again, giving AL no advantage.

AL is a promising field and its application in the domain of AAR is novel, having particular importance due to the extra challenges of animal activity annotation as discussed in the paper. As can be seen, random sampling already performs very well and sometimes equally well as AL. However, AL allows performing smarter and more efficient labelling of instances, which are more important than others in the training dataset, when classifying many activities at a low F1 score. This accelerates the training process, reducing the manual effort needed by annotators, which are hard to find and expensive in the case of monitoring wildlife.

References

1. L. Gao, H. A. Campbell, O. R. Bidder, and J. Hunter, "A Webbased semantic tagging and activity recognition system for species' accelerometry data," *Ecological Informatics*, vol. 13, pp. 47–56, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ecoinf.2012.09.003

2. V. Sturm, D. Efrosinin, N. Efrosinina, L. Roland, M. Iwersen, M. Drillich, and W. Auer, "A Chaos Theoretic Approach to Animal Activity Recognition," *Journal of Mathematical Sciences (United States)*, vol. 237, no. 5, pp. 730–743, 2019.

3. J. W. Kamminga, D. V. Le, J. P. Meijers, H. Bisby, N. Meratnia, and P. J. Havinga, "Robust Sensor-Orientation-Independent Feature Selection for Animal Activity Recognition on Collar Tags," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies IMWUT*, vol. 2, no. 1, pp. 1–27, 2018. [Online]. Available: https://dl.acm.org/citation.cfm?id=3191747

4. B. Settles, *Active Learning, Burr Settles*, 2013.

5. N. V. Cuong, W. S. Lee, and N. Ye, “Near-optimal adaptive pool-based active learning with general loss,” *Uncertainty in Artificial Intelligence - Proceedings of the 30th Conference, UAI 2014*, pp. 122–131, 2014.

6. A. Vaith, “Uncertainty based active learning with deep neural networks for inertial gait analysis,” *FUSION*, vol. 23, p. 8, 2020.

7. P. Ren, Y. Xiao, X. Chang, P. Y. Huang, Z. Li, X. Chen, and X. Wang, “A survey of deep active learning,” *arXiv*, 2020.

8. M. Stikic, K. Van Laerhoven, and B. Schiele, “Exploring semisupervised and active learning for activity recognition,” *Proceedings International Symposium on Wearable Computers, ISWC*, pp. 81–88, 2008.

9. R. Liu, T. Chen, and L. Huang, “Research on human activity recognition based on active learning,” *2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010*, vol. 1, no. July, pp. 285–290, 2010.

10. I. Muslea, I. Muslea, S. Minton, S. Minton, C. a. Knoblock, and C. Knoblock, “Selective sampling with redundant views,” *Proceedings of the National Conference on Artificial Intelligence*, p. 621–626, 2000. [Online]. Available: http://www.aaai.org/Papers/AAAI/2000/AAAI00095.pdf

11. M. Dredze and K. Crammer, “Active Learning with Confidence 2008,” no. June, pp. 233–236, 2008.

12. J. Zhu, H. Wang, B. K. Tsou, and M. Ma, “Active learning with sampling by uncertainty and density for data annotations,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1323–1331, 2010.

13. W. Di and M. M. Crawford, “View generation for multiview maximum disagreement based active learning for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 5 PART 2, pp. 1942–1954, 2012.

14. J. W. Kamminga, L. M. Janßen, N. Meratnia, and P. J. M. Havinga, “Horsing Around—A Dataset Comprising Horse Movement,” *Data*, vol. 4, no. 4, p. 131, 9 2019. [Online]. Available: https://www.mdpi.com/2306-5729/4/4/131

15. L. Gulf Coast Data Concepts, “Human Activity Monitor: HAM,” online, 2019. [Online]. Available: http://www.gcdataconcepts.com/ham.html

16. J. Kamminga, “Matlab movement data labeling tool,” 8 2019.

17. A. Olszowka-Myalska and J. Chrapo´ nskib, “Active Learning with Real´ Annotation Costs Burr,” *Solid State Phenomena*, vol. 227, pp. 178–181, 2015.

18. M. Lorbach, R. Poppe, and R. C. Veltkamp, “Interactive rodent behavior annotation in video using active learning,” *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 19787–19806, 2019.

19. R. G. Jansen, L. F. Wiertz, E. S. Meyer, and L. P. Noldus, “Reliability analysis of observational data: Problems, solutions, and software implementation,” *Behavior Research Methods, Instruments, and Computers*, vol. 35, no. 3, pp. 391–399, 2003.

20. D. D. Lewis and W. A. Gale, “A sequential algorithm for training text classifiers,” *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, pp. 3–12, 1994.

21. L. Copa, D. Tuia, M. Volpi, and M. Kanevski, “Unbiased query-bybagging active learning for VHR image classification,” *Image and Signal Processing for Remote Sensing XVI*, vol. 7830, no. October 2010, p. 78300K, 2010.