BlobSnake: Gamification of Feature Extraction for ‘Plug and Play’ Human Activity Recognition.

Reuben Kirkham[a], Carlton Shepherd[b] & Thomas Plötz[a]
OpenLab, Newcastle University, Newcastle upon Tyne, UK[a]
Information Security Group, Royal Holloway, University of London, London, UK[b]
r.kirkham@newcastle.ac.uk

ABSTRACT

We present BlobSnake, a casual game designed to help generate new feature representations in the context of Human Activity Recognition. Feature selection is an essential task to be completed in the context of developing any non-trivial activity recognition system for a new set of activities. Presently, using anything other than a set of standard features requires a considerable amount of effort to be expended upon expert driven algorithm development. BlobSnake is an alternative approach which uses direct interaction with real sensor data by non-experts in order to develop additional features, thus lowering the cost and expertise otherwise required to produce more effective recognition performance. Our experiments demonstrate that our method improves upon the state of the art performance of standard features in a challenging recognition scenario.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Crowdsourcing; Gameification; Human Activity Recognition; Machine Learning

INTRODUCTION

The field of wearable computing has grown over the past decade to now encompass systems that automatically recognise human behaviour in real time. This progress has arisen from developments in the field of Human Activity Recognition (HAR). HAR is the process of transforming sequential data, primarily from inertial sensors (e.g accelerometers and gyroscopes), into an indication of which specific type of activity a person is undertaking at a given period in time. Over the course of the past decade, HAR has progressed from the detection and classification of very simple ambulatory motion (determining between walking, standing still, and running) [3] onto an rapidly increased number of activities and special user-groups, including specific disabilities (e.g. [2, 25, 1, 12]), and even animal based activities [22, 16, 29].

British HCI 2015, July 13 - 17, 2015, Lincoln, United Kingdom.
Copyright © 2015 ACM ISBN/978-1-4503-3643-7/15/07 $15.00.
http://dx.doi.org/10.1145/2783446.2783579

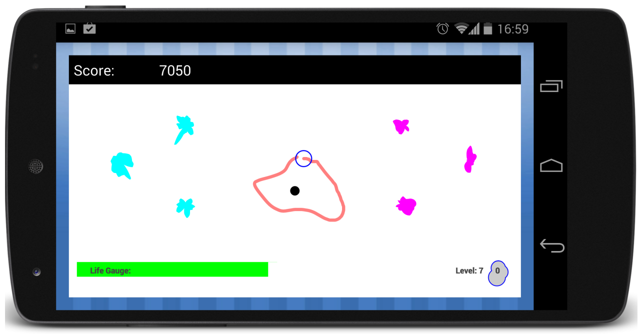

Figure 1. The BlobSnake game interface. The player draws a shape which is designed to represent the group of blobs on the left better than those on the right. Each colour of blob is drawn from a different class, whilst the blobs themselves are transformed deterministically from a given channel of raw sensor data.

The technical development of a new HAR system is often a burdensome endeavour, especially in cases where a high recognition accuracy is required. Asides certain relatively trivial recognition problems, a scenario arising from a specific activity dataset usually requires the tailored development and training of a HAR system. The core parts of this process involves collecting and annotating a activity dataset, before an HAR expert then develops a specific feature representation that will work effectively for that given problem. This work is focussed upon the latter challenge - i.e. developing this specific feature representation.[1]

In the near future, we will require an ever increasing number of HAR systems which focus upon more technically challenging HAR problems. The context of disability represents a particularly iconic illustration of why this is so, in that it presents the need to develop a broad range of new systems which recognize activities that relate to specific actions related to a given disability (e.g. as in [10]), which in turn demands a wider range of trained recognition systems. Moreover, there is now a renewed emphasis upon recognising individual gestures, which are particularly challenging activities to recognise due to the short time period

[1]For an investigation aimed towards addressing the former concern through annotation correction, see [20].

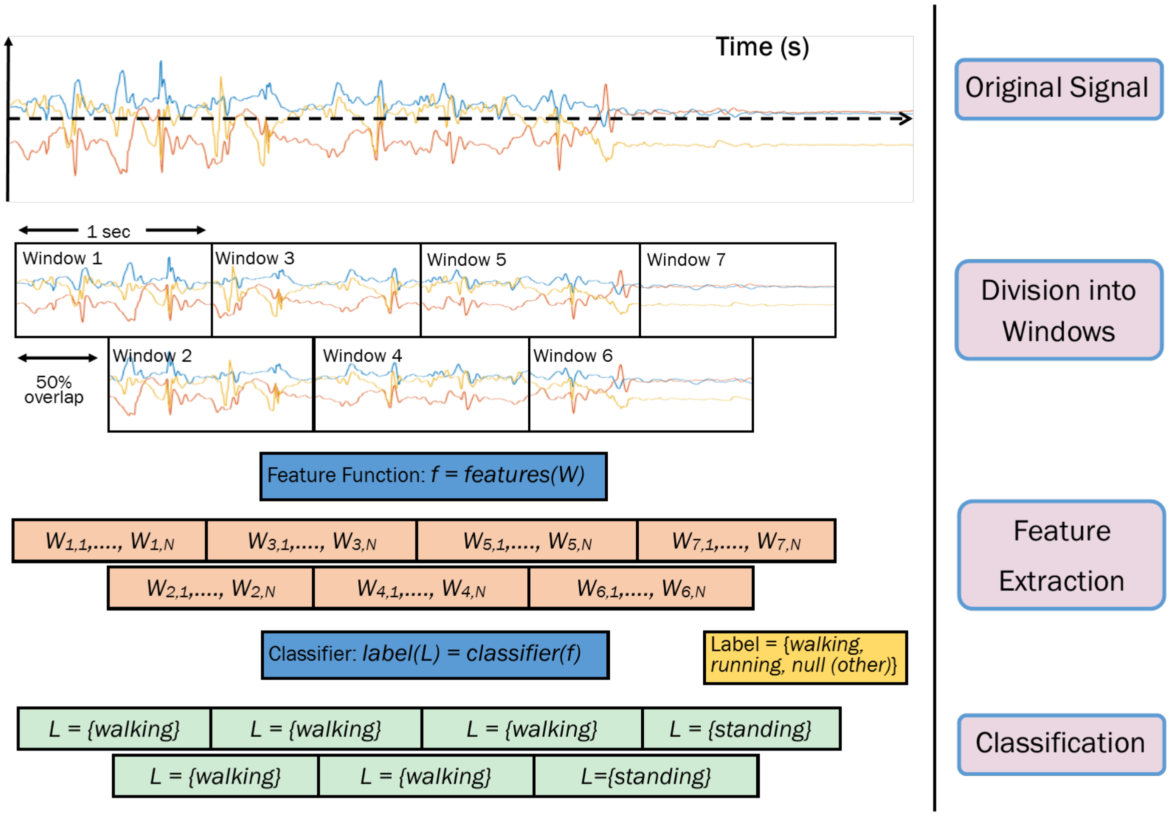

Figure 2. The standard 'sliding window' approach used for human activity recognition. This begins with sensor data, which is partitioned into windows (typically these are 1 second long and with a 50% overlap, although this can vary in different application contexts). Each window is fed into the feature function in turn. The feature representation that results is then fed into the classifier which returns a label (which is the specific activity the system thinks the user is currently doing) for each window.

in which they actually occur (with events often lasting a fraction of a second) and the more subtle distinctions between activity classes [7].

BlobSnake (**Figure 1**.) is a casual mobile game which has been developed in order to assist in creating new feature representations. The game mechanic is based upon drawing abstract shapes representing the raw sensor data depicted in the game itself. Our innovation is the development of a performance metric which can be used to both motivate gameplay and serve as an effective feature for improving activity recognition performance. Importantly, this allows lay users to effectively replace existing HAR experts, thereby enabling a broader range of HAR systems to be developed (within existing economic constraints).

There already exist 'plug and play' feature representations which do not require the use of a HAR expert, most notably the ECDF [11] when used as part of a standard sliding window recognition pipeline. We show that BlobSnake is a means towards improving upon the performance of these standard features, using the challenging Opportunity Dataset to demonstrate a substantive improvement in recognition performance [7]. The result is a small step towards moving HAR on from being a purely expert activity, onto a participatory one in which the general public becomes involved as part of the process of developing new HAR systems.

BACKGROUND AND RELATED WORK

The Process of Human Activity Recognition

In order to understand the rational behind BlobSnake, we explain the anatomy of a sensor-based Human Activity Recognition system. HAR generally involves the adaption of traditional machine learning algorithms so that they can be applied to sequential sensor data. In practise, this usually means deriving a feature representation and extraction framework, before then applying this to a static classifier. In this setting (there are alternative approaches towards developing a HAR system, although for practical reasons we focus upon the more usual approach we outline in **Figure 2**) the main focus and challenge is developing a translation of the data into a suitable form for one of these classifiers.

Most HAR systems rely upon a sliding window approach (overviewed in **Figure 2**), where features are computed using a standard framework for a fixed window in time (normally a matter of a few seconds), which 'slides' along the data. In a deployed HAR system, there are four stages: collection of a signal, its subsequent division into windows, feature computation and classification into a specific set of activities. The signal would usually come from one or more inertial sensors (accelerometers and gyroscopes), before

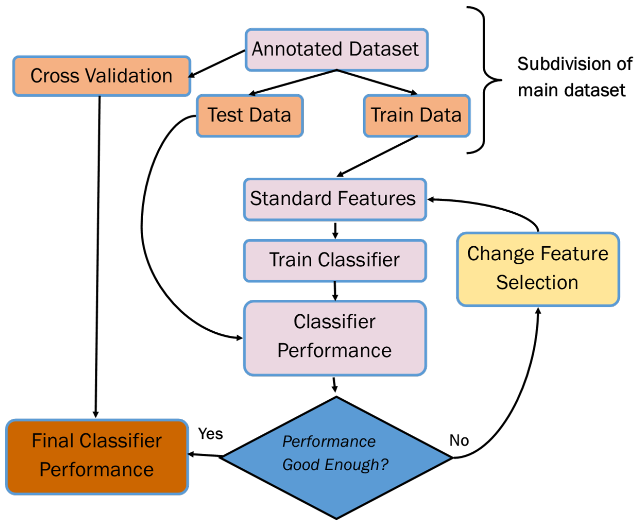

Figure 3. An overview of the core development process widely used for creating a new HAR system. First the dataset is usually divided into three datasets, one used to train the classifier, and two to evaluate the recognition performance of a system. The developer then iteratively develops and trials a range of feature representations, with a selection of classifiers, noting the recognition performance each time on the test set. Once the desired recognition performance is reached, the given feature representation and classifier combination is selected, before being evaluated on a separate cross-validation dataset to reach a final recognition performance figure.

being divided into windows. Each window would then usually have a deterministic feature function applied to it, with the results then being fed into a standard classifier.

BlobSnake is designed to assist with one part of the technical development of a HAR system, namely the development of a feature representation function. Indeed, once a dataset has been collected and annotated, and a sliding window approach has been adopted, there still remains the need for a feature representation to be developed, making BlobSnake potentially useful in these cases, too, then feature representation is the core technical task that needs to be completed by a HAR expert, because it would be unreasonable (and impractical) to develop a new classifier for each given HAR problem. **Figure 3** shows the flow of this process, and the general pattern followed by HAR experts in developing their systems.

Gameification and Crowdsourcing

BlobSnake is not the first effort aimed at gamefying (or indeed crowdsourcing) Human Intelligence Tasks. The most prominant examples are Galaxy Zoo [6] which focusses upon pattern identification in Astronomy, and Fold-It [8], which applies gameification to protein folding, and suggests solutions for future evaluation for researchers. Both platforms have users which number in their millions, and have produced a significant volume of real world results. Perhaps most importantly, they step beyond crowdsourcing, having widely engaged the general public as citizen scientists, who not only help make meaningful contributions to science, but also have the opportunity to understand and become enthused by scientific research as an end in and of itself.

Some perhaps lesser known efforts focus upon the Machine Learning context. These range from works designed to improve search algorithms [21], onto efforts that involve interaction with artificial chemistries [14]. Von Ahn and Dabbish [30] go as far as to provide detailed design guidance for systems in this space, which emphasises the benefits of gameification, moving from a mechanised approach to one that actually engages the ’crowd-worker’. These works have also had an impact in engaging amateur scientists, albeit on a smaller scale relative to their more well-known counterparts.

There have been some efforts which have aimed at providing an improved HAR system [23, 24]. However, these are limited in that they are not aimed at developing a final recognition system, but instead use the crowd to partially replace it in real time. By contrast, BlobSnake is focussed upon helping to develop a system that can subsequently be entirely automated, and thus can be deployed efficiently in a wide range of automated contexts. Neither does Blobsnake require a large crowd (and a live internet connection to connect a user to them), or indeed a crowd at all - because of the small size of HAR datasets - the existing system only requires a few hours of human interaction. This has an advantage of allowing its use without any difficulty in existing ethics regimes, because there is no need to release the data to a ‘public crowd’. Thus, BlobSnake should be thought of as an engaging interface that can be used to assist in developing new HAR systems.

Gameification is naturally founded upon creating genuinely enjoyable gaming experiences. As Huizinga notes [13], why would you play a game if it is not fun? The foundational book by Salen and Zimmerman [27] provides a detailed and longstanding set of design patterns and insights widely used within the games development community. This is complemented by a wide range of literature which explores aspects of play. Juul [17] provides an engaging exploration of how the separation between the interface (or controls) used for a gaming experience and the gameplay itself is an artificial distinction, and provides suggestions upon how to make the controls an integral part of the gaming experience instead. Cairns et al [5] provide an overview of immersion - which might be loosely described as engagment - and the research over the previous decade explaining how this varies depending upon certain elements of a gaming experience. All of this research, was drawn upon when designing BlobSnake, with an emphasis upon ensuring that the gaming experiences is as enjoyable as possible.

GAMEPLAY

Before explaining the underlying algorithm, it is worth (briefly) overviewing the gameplay of BlobSnake and some of its salient design features. An overview of the user interface can be found in **Figure 4**. The core idea is that player is presented with a number of blobs, each of which represents a single channel from a window of sensor data (see **Figure 5** for how they are generated). The left hand side (and colour) represents one group, whereas the right hand side (with a different colour) represents another. The goal is for

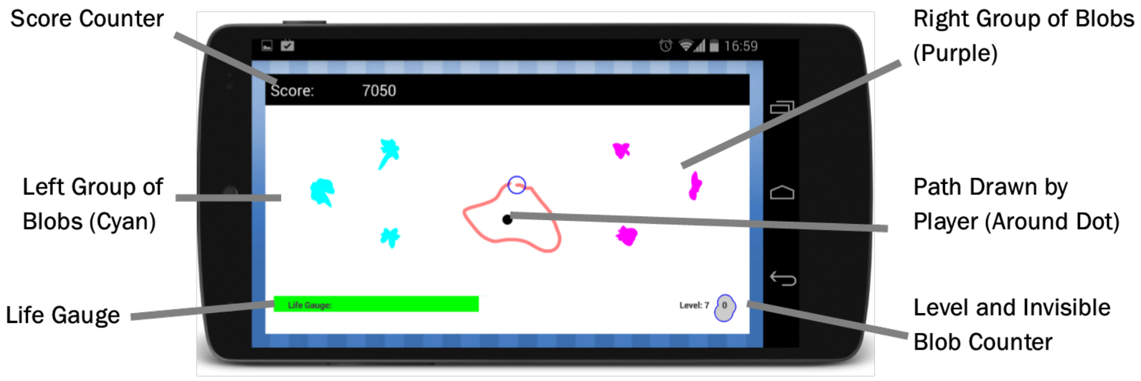
Score Counter
Left Group of Blobs (Cyan)
Life Gauge
Right Group of Blobs (Purple)
Path Drawn by Player (Around Dot)
Level and Invisible Blob Counter
Score: 7050
Life Gauge:
Level: 7
0
16:59

Figure 4. An overview of the core gameplay features in BlobSnake

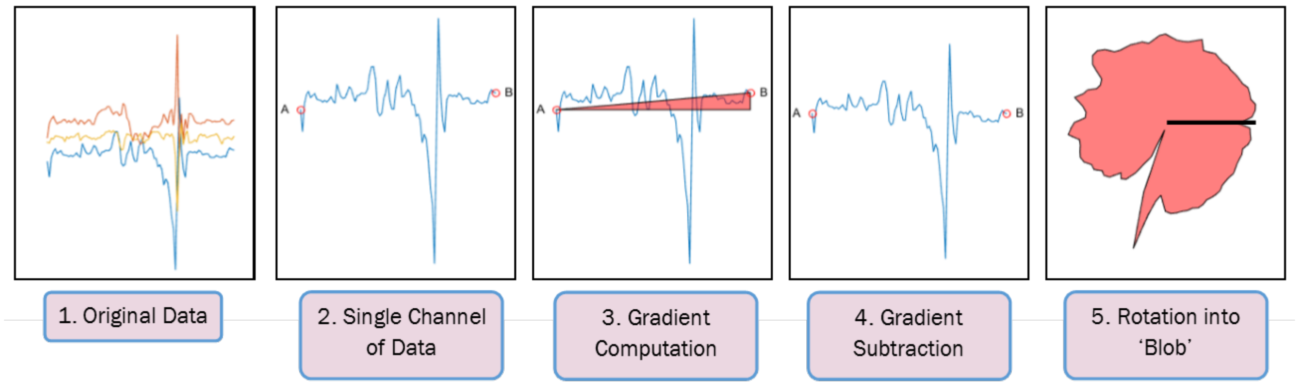

Figure 5. The stages of mapping a from a window of data (1). The process works on a channel by channel basis (2). In (3) a gradient is computed before being added to the data (4) in order to make the last sample equal to the first sample. The data is then transformed by a computed scale factor in Eq. 1. (not shown), before being rotated to join end to end (5) where the black line indicated where A and B (from (4)) have been joined together.

the player to draw a shape around the central dot that fits one group of blobs better than the other.

There are some significant game play elements (the full technical detail is provided later within this paper) which help create a sense of engagement on behalf of the player. First, there is a fitting algorithm that is applied to the shape and each individual blob in turn. The average difference in fit between the two groups is computed; the greater this is, the more the score is increased by. Second, the game also includes a life bar which increases or decreases based upon a combination of the time taken (over a minimum threshold) to draw a blob and the result of that fitting algorithm.

BlobSnake draws upon the notions of challenge and progression, which are generally recognised as being necessary for facilitating effective gameplay [27]. Some of the challenge is within the controls of the game (adopting Juul's tactic here of making the controls an integral part of the gameplay experience [17]), namely the task of drawing a shape (which appears as it is drawn) in the confined space presented by BlobSnake. From the perspective of progression, BlobSnake also involves an increasing number of blobs displayed on each side, and eventually there are invisible blobs which are taken into account in the 'difference of fit' calculation and also scored. In addition, there is the usual (for casual games) score-board included to foster a sense of competitiveness amongst players, with the goal of motivating continued play.

These gameplay elements are not only chosen to create an engaging gaming experience. They are also included to promote behavior that is likely to lead to effective features being generated through the game. In machine learning (including HAR), there are the concepts of *overfitting* and *underfitting*. These effectively correspond to producing results that too closely mimic the training data and thus do not generalise (i.e. overfitting) or alternatively, have little closeness to the data itself (i.e. underfitting) and thus represents little useful information that can be used for classification. In practice the challenge with BlobSnake is overfitting rather than underfitting, because players might adopt a strategy of directly drawing one of the blobs. The time limiting function, together with the small screen size, are designed to minimize this problem. Moreover, the fitting metric should have the effect of limiting underfitting, because this is likely to reduce the likelihood of the player producing a poor fit (in that they will be penalised in both life-bar levels and score for doing so).

BLOBSNAKE ALGORITHMS

For the game itself, there are two key technical aspects. The first is how the sensor data is mapped into the blobs that

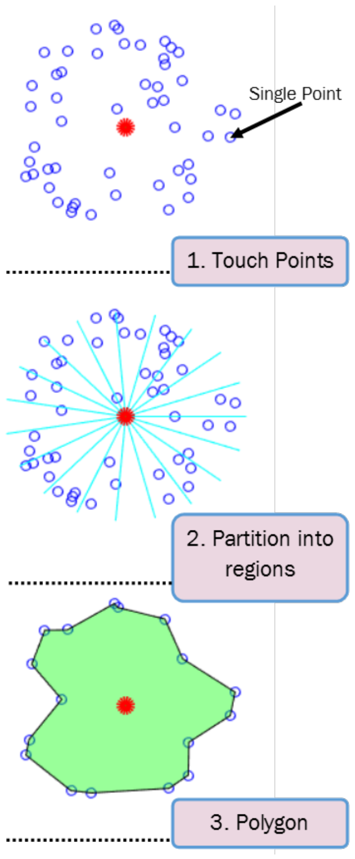

Figure 6. This figure demonstrates the conversion of data to polygons. Starting from individual touch points (1), onto selcting equally spaced particles moving out from a central point (2), producing a polygon (3). Note that this process is the same as taking the convex hull of the point cloud.

appear in the game itself, from an individual window and channel of sensor data. The second is how the fitting energies - which we use to drive the scoring mechanism in the game itself - are calculated and used in order to determine the efficacy of the 'snake' drawn by the player. Both of these are explained below. As will be seen later in this article, these will also be fundamental components of the HAR process, with identical calculations being used at the feature selection stage.

Generating Blobs

The first core technical consideration is how to map real sensor data meaningfully into the gamespace as a 'blob'. This is partially based upon our previous work [19], which presented activity data directly to users and found a circular approach to signal representation was effective as a means of communicating this data in a meaningful way. We implement this using a linear transform illustrated into **Figure 5** in order to make the beginning and the end of the recognition data join smoothly to form a 'blob'. That is, for each datapoint p_i (where $p_A p_0$ is the first point of that window, p_B is the end point, and N is the number of points in the blob), we compute the new value p_i':

$$p_i' = p_i - \frac{i(p_B - p_A)}{N - 1} \tag{1}$$

The data frame F (this is equivalent to a single channel - or axis - of sensor data from a sliding window) is then scaled using the following scale factor (where $\Delta F = |F_{min} - F_{max}|$):

$$SF = \begin{cases} 1/|\Delta F| & \text{if} |\Delta F| \leq 2 \\ \log(1/|\Delta F|) & \text{otherwise} \end{cases} \tag{2}$$

This formula has the advantage of removing the mean (so our players do not reproduce this as a feature), whilst preventing signal noise being greatly amplified as to appear meaningful.

Evaluating Snakes in Real Time

In computer vision, a snake [18] is a technique which uses fitting energies in order to detect shapes, with the aim of minimizing the fitting energy. We adapt this concept to measure how faithfully a 'snake' from our game fits an individual blob, with the 'snake' being instead the shape drawn by the player. First, the touch inputs are processed by finding the values closest to the central dot (see **Figure 1.**), with a point generated for each segment that corresponds to a data point (in our case we take 1 point for every 12 degrees, making 30 in total). We illustrate in **Figure 6** the first step of the process, converting touch points into a polygon.

The blob is actually comprised of a discrete set of points, or samples, in the sensor data. Thus, both the blob and the snake are polygons of the same size (or can be made so by simple interpolation on the blob). We therefore measure the deformation by normalising each polygon to have a length of one, and then computing the difference in each segments length. This provides features that can be used for a classifier, where each snake is effectively deformed to fit a 'blob'. The relative segment lengths for the original snake are computed, before the energy being computed by summing each segment's deformation. This is computed for every possible rotation of the snake (relative to the blob), with the minimum value selected as the final fitting energy. Mathematically this is defined in the following equation:

$$E = \min_{S_r} \sum_{seg=1}^{n} abs\left(\frac{L_{seg}}{\sum L_{seg}} - \frac{L'_{seg}}{\sum L'_{seg}}\right) \tag{3}$$

In the above, S_r is the snake rotated to begin at point r, L_{seg} is the original segment length and L'_{seg} is the segment length after the snake has been fitted to the blob (which is always equal to the relevant segment length in the blob itself).

TECHNICAL EVALUATION

Having detailed the game and how snakes and blobs are generated, the next step is to consider whether this can be usefully utilsed in a HAR context. Given this submission's focus upon the interface, we do not emphasise the development of novel algorithms in this evaluation, instead opting for a simple filtering and grid search approach (detailed below).

From Snakes to Features

We draw upon the blob generation and snake fitting metrics defined in the game itself. First, every data-frame from each channel (in the dataset) is converted into a blob, and each snake is evaluated across all the blobs for each class using exactly the same method as above. Thus we re-use the fitting energy from the game metric, but this time the indivdiual snake is a feature function, applied to channels of sensor data first transformed into blobs. Because there is a likelihood of a large amount of erroneous solutions generated by our users, we adopt the approach from FoldIt [8] of initially scoring user inputs, taking only the most promising generated 'snakes' forward in order to reduce our dataset. To do this, we conduct the below calculation for each snake s in turn, where μ_c and σ_c are respectively the mean and standard deviation of class c and then selecting those N snakes which are the highest scoring to further evaluate:

$$s_{score} = \frac{abs(\mu_1 - \mu_2)}{\sigma_1 + \sigma_2} \tag{4}$$

Note that this function is defined to maximize the separability of the snake with respect to different classes (and thus the usefulness of a given snake as a feature), by looking the for the highest difference in means, whilst normalizing for standard deviation.

We then presume that the best performing standard feature representation (out of those traditionally used) has already been identified for our data. This is accomplished by using a standard framework as described in [4], with this being trialled on a wide range of classifiers and standard feature representations. The goal of BlobSnake, just as with the HAR expert in the traditional approach, is to add complementary features that improve upon the standard features, rather than to replace the standard features. Thus before our experiments, the best performing standard features were identified - this was the ECDF with 10 coefficients (more co-efficients did not improve - or substantively reduce - recognition performance). A grid search optimisation is then performed on the training data to determine which snakes offer the best raw performance (in accord with the formula above) to take forwards, by concatenating them individually - to the highest performing (standard) feature representation. If there are two or more features which individually improve performance across different sensor channels that also exceed a given threshold, T, then these combinations are also trialled. The final selection is the one with the greatest recognition performance.

Dataset

For our evaluation, we use the challenging Opportunity Dataset [7], which is one of the most widely recognised datasets for benchmarking HAR systems. Opportunity comprises a number of short time activities in a domestic environment, most of which are mundane gestural actions, such as opening a specific door (part of the associated challenge with this dataset involves distinguishing the use of one door from another), or activating a toggle switch. Notably, this is the only dataset which is currently publicly available which is also suitably challenging (and large enough to enable an appropriate evaluation); other datasets already obtain very high recognition performance using standard features, and thus BlobSnake would have little utility in that context. Ultimately, BlobSnake is forward looking, and aimed at the more challenging HAR problems which are beginning to arise going forwards, and thus it would make little sense to trial it on problems that are already solved.

The Experiments

We selected the five most challenging class pairs in Opportunity to form five separate experiments. The goal was to demonstrate the viability of our approach against an extremely challenging HAR problem. These class pairs were identified by the highest pairwise confusion under the standard sliding window framework, with the usual 1 second windows and 50% overlap (this was established as the default approach in [15]). In order to identify the best performing standard feature representation, we trialed 1NN, 3NN, C4.5 and Naïve Bayes as classifiers, against the standard statistical features, ECDF [11], PCA, and Fourier coefficients in turn (with 5,10,15,20,25 and 30co-efficients) on the Opportunity dataset in totality.

This resulted in 45 individual blob-snake games (our configuration of Opportunity has 9 sensor channels, because we focus upon the three accelerometer sensors relating to the dominant wrist). The case with 1NN and the ECDF representation provided the greatest recognition performance, and the improvement saturated with 10 co-efficients per channel, which then served as our basis for expanding upon. For our experiments 50% of the data from each class pair were randomly selected to be deployed within the game as a training dataset, with the players never seeing any of the test data to ensure a robust evaluation. Because there is effectively only one system on trial to be deployed (BlobSnake), we dispense with the usual cross-validation dataset that we would otherwise would have been forced to use.

We then recruited eight participants to play BlobSnake, all of whom were 'naïve' with respect to HAR. Our participants were recruited via a student volunteer mailing list and offered an inducement of £10. They were asked to play the game for a period of around 45 minutes, with a brief explanation offered at the outset of how the game operated (including a tutorial in the game itself), in order to generate user data (and thus 'snakes'). After the 'snakes' were collated in totality from the laboratory studies, they were processed using the post-processing framework highlighted in the foregoing section (with a threshold of $T = 0.5\%$ and filtering at the level of $N = 10$ as described above).

Results

We report the results in **Table 1**, with the McNemar test used with the appropriate statistical corrections to this context (Yates and Bonferroni), as an improvement against 1NN with 10 ECDF coefficients (namely the best possible performance with standard features). The numerical results are comparable with other research efforts (e.g. [26, 11]) designed to improve upon state of the art recognition systems in the HAR context. Unfortunately, these papers do not normally follow up with a

Experiment No	Class A	Class B	Original Performance (% Accuracy)	Improved Performance (% Accuracy)	Total Improvement (% Accuracy)	P-value
Experiment 1	Toggle Switch	Open Dishwasher	67.24	68.10	0.86	0.241
Experiment 2	Close Dishwasher	Close Fridge	72.34	72.99	0.65	0.343
Experiment 3	Open Door 2	Close Drawer 3	73.90	74.48	0.58	0.422
Experiment 4	Open Drawer 1	Open Drawer 2	73.97	74.19	0.22	0.683
Experiment 5	Close Drawer 1	Open Door 1	76.99	78.87	1.88	0.021

Table 1. The results from Blob Snake; with a numerical improvement in all experiments beyond the *state of the art*, and a statistically significant improvement in Experiment 5 (even with Bonferroni Correction). Notice that this improvement is in the usual range for new approaches towards HAR, making this a substantial development in this field.

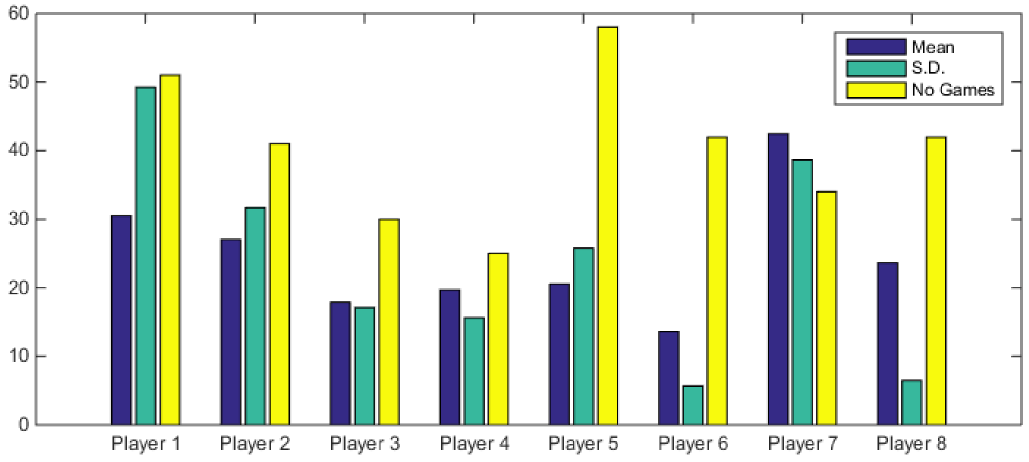

Figure 7. Player statistics. For each player, we provide the mean and standard deviation of the number of blobs generated within the game, as well as the total number of games played by individual players. As can be seen, these vary substantially between individual players, suggesting different approaches towards gameplay, and player skill across our players.

statistical analysis (e.g. [28]), so we cannot easily compare with them in that regard.

In **Figure 7** we also report the statistics in relation to the gameplay for each player. As can be seen, there is a great degree of variance in respect of both playstyle and ability, given the number of snakes generated, and the standard deviations that results. This is in effect what the game was designed to achieve, with the gameplay features being selected to encourage a diversity in play-style, and thus the data generate. It is also notable that it is unlikely that further play would improve performance, both due to the number of snakes generated (in excess of 8000), and the diversity within that data.

DISCUSSION AND CONCLUSION

Can human interaction with a casual game displace the efforts of HAR experts? With BlobSnake, we demonstrate that this is likely to be possible to accomplish in a meaningful fashion. In our experiments, we found that BlobSnake successfully improves activity recognition performance, doing so in a statistically significant fashion for one data-pair (and numerically for others) on a highly challenging HAR dataset. In doing so, we have stepped towards reducing the number of systems that require active development, whilst opening up a new domain for research for the HAR community.

This work was in effect a feasibility study for this approach, which we hope to expand upon in further work. As such, there is a wide range of further steps that might be trialed in order to create an improved system. This might include the development of more powerful post-facto algorithm which better makes use of the snake data, an investigation of different means of data representation, and thus different ‘blobs’ (e.g. representing multiple channels of sensor data in a single artifact, or using time delay embeddings [9]). Further work may also involve taking into account play characteristics (for example, the time taken to generate each blob by the player, or the player’s current score) when selecting which blobs are used.

HAR systems increasingly represent challenges involving privacy, and it is important that this issue is effectively explored, so people can make informed decisions about how and where they use wearable systems that contain inertial sensors. This is an important avenue for further research which we also hope to explore in the future.

REFERENCES

1. Albinali, F., Goodwin, M. S., and Intille, S. Detecting stereotypical motor movements in the classroom using accelerometry and pattern recognition algorithms. *Pervasive and Mobile Computing 8*, 1 (Feb. 2012), 103–114.
2. Bachlin, M., et al. Wearable assistant for Parkinson’s disease patients with the freezing of gait symptom. *IEEE Trans. Information Technology in Biomedicine 14*, 2 (2010).

3. Bao, L., and Intille, S. S. Activity recognition from user-annotated acceleration data. In *Pervasive* (2004).

4. Bulling, A., Blanke, U., and Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys* (2013).

5. Cairns, P., Cox, A., and Nordin, A. I. Immersion in digital games: a review of gaming experience research. *In Handbook of Digital Games (eds M. C. Angelides and H. Agius)* (2014).

6. Cardamone, C., et al. Galaxy Zoo Green Peas: discovery of a class of compact extremely star-forming galaxies. *Monthly Notices of the Royal Astronomical Society* (2009).

7. Chavarriaga, R., and et. al. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* (Jan. 2013).

8. Eiben, C., and et. al. Increased Diels-Alderase activity through backbone remodeling guided by Foldit players. *Nature Biotechnology. 30*, 2 (2012).

9. Frank, J. Activity and Gait Recognition with Time-Delay Embeddings Time-Delay Embeddings. In *Proceedings of AAAI* (2010).

10. Goodwin, M., Intille, S., Albinali, F., and Velicer, W. Automated detection of stereotypical motor movements. *Journal of Autism and Developmental Disorders 41*, 6 (2011), 770–782.

11. Hammerla, N., et al. On Preserving Statistical Characteristics of Accelerometry Data using their Empirical Cumulative Distribution. In *ISWC* (2013).

12. Hammerla, N., et al. Pd disease state assessment in naturalistic environments using deep learning. AAAI 2015 (2015).

13. Huizinga, J. Nature and significance of play as a cultural phenomenon. *The performance studies reader* (2004), 117–20.

14. Hutton, T. The Organic Builder: A Public Experiment in Artificial Chemistries and Self-Replication. *Artificial Life* (2009).

15. Huynh, T., and Schiele, B. Analyzing Features for Activity Recognition. *Joint Conference on Smart Objects and Ambient intelligence* (2005).

16. Jackson, M. M., et al. Fido - facilitating interactions for dogs with occupations: Wearable dog-activated interfaces. ISWC 2013 (2013).

17. Juul, J., and Norton, M. Easy to Use and Incredibly Difficult : On the Mythical Border between Interface and Gameplay. In *International Conference on Foundations of Digital Games* (2009).

18. Kass, M., Witkin, A., and Terzopoulos, D. Snakes : Active Contour Models. *International Journal of Computer Vision 331* (1988).

19. Kirkham, R., et al. The Breaktime Barometer An Exploratory System for Workplace Break-time Social Awareness. In *Ubicomp* (2013).

20. Kirkham, R., Khan, A., Bhattacharyra, S., Hammerla, N., Mellor, S., Roggen, D., and Plötz, T. Automatic Correction of Annotation Boundaries in Activity Datasets by Class Separation Maximization. In *Ubicomp 2013 Adjunct* (2013).

21. Klau, G., Lesh, N., Marks, J., and Mitzenmacher, M. Human Guided Search. *Journal of Heuristics* (2010).

22. Ladha, C., et al. Dogs Life: Wearable Activity Recognition for Dogs . In *Proc. of. Ubiquitous Computing* (2013).

23. Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. Real-time crowd labeling for deployable activity recognition. *Proc. of CSCW* (2013).

24. Lasecki, W. S., Weingard, L., Ferguson, G., and Bigham, J. P. Finding Dependencies Between Actions Using the Crowd. In *Proc. of CHI* (2014).

25. Plötz, T., et al. Automatic Assessment of Problem Behavior in Individuals with Developmental Disabilities. In *Ubiquitous Computing 2012* (2012).

26. Reiss, A., Hendeby, G., and Stricker, D. Confidence-based multiclass adaboost for physical activity monitoring. ISWC '13 (2013).

27. Salen, K., and Zimmerman, E. *Rules of play: game design fundamentals*. 2004.

28. Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., and Tröster, G. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing 7*, 2 (2008), 42–50.

29. Thompson, R., et al. Dancing with horses: Automatic quality feedback for dressage riders, to appear in proc. ubicomp. Ubicomp 2015 (2015).

30. von Ahn, L., and Dabbish, L. Designing games with a purpose. *Communications of the ACM 51*, 8 (Aug. 2008).