World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:17, No:11, 2023

Improving Activity Recognition Classification of Repetitious Beginner Swimming Using a 2-Step Peak/Valley Segmentation Method with Smoothing and Resampling for Machine Learning

Larry Powell, Seth Polsley, Drew Casey, Tracy Hammond

Abstract—Human activity recognition (HAR) systems have shown positive performance when recognizing repetitive activities like walking, running, and sleeping. Water-based activities are a reasonably new area for activity recognition. However, water-based activity recognition has largely focused on supporting the elite and competitive swimming population, which already has amazing coordination and proper form. Beginner swimmers are not perfect, and activity recognition needs to support the individual motions to help beginners. Activity recognition algorithms are traditionally built around short segments of timed sensor data. Using a time window input can cause performance issues in the machine learning model. The window's size can be too small or large, requiring careful tuning and precise data segmentation. In this work, we present a method that uses a time window as the initial segmentation, then separates the data based on the change in the sensor value. Our system uses a multi-phase segmentation method that pulls all peaks and valleys for each axis of an accelerometer placed on the swimmer's lower back. This results in high recognition performance using leave-one-subject-out validation on our study with 20 beginner swimmers, with our model optimized from our final dataset resulting in an F-Score of 0.95.

Keywords—Time window, peak/valley segmentation, feature extraction, beginner swimming, activity recognition.

I. Introduction

The availability of commercially available, waterproof devices has facilitated Human Activity Recognition systems (HAR) to detect and classify water activities like swimming and have moved away from just land-based activities. However, researchers and device manufactures focus on Olympic lap swimming activities (backstroke, breaststroke, butterfly, and freestyle) and do not adequately cover beginner or novice swimmers. Commercial devices, such as Fitbit, Apple Watch, and Garmin watches, work well for elite and professional swimmers but are not effective in less regimented swimming activities. Morais et al. studied the classification of commercial devices and found that the accuracy decreases significantly when used by beginner and intermediate swimmers [1]. Our research looked at implementing a method of detecting swimming activity of all levels by using a two-step segmentation method.

HAR research is a field that uses sensors, algorithms, and wearable systems to identify physical activities. Researchers have been able to integrate activity recognition in such areas as sports (e.g., running) [2], hobbies (e.g., playing piano) [3], and activities of daily living (e.g., eating, taking pills, brushing teeth) [4]. Activity recognition in these papers works by segmenting the sensor data based on a window of time and using the data for classification. However, determining the proper time window is challenging and is a major effect on the accuracy of the classification algorithm [5].

Selecting the appropriate time window can be difficult when considering the variety of users and can be a significant factor in the classification accuracy of the machine learning algorithm [6]. If the system uses a short window, there may not be enough information to grasp the activity performed. Still, if it was too long of a window, the person could have performed multiple activities within that time [5]. Algorithms can misclassify data due to the lack of fidelity in the data from a misaligned time window. Finding the perfect length for recognizing the desired activity is complex and requires a lot of trial and error. However, using a time window to segment the data is currently the proper method to get a consistent data flow for classification. Though multiple activities exist, the window must fit the overall activity perfectly. There needs to be another method that dynamically segments the data, and it should not rely on a perfect static time window.

In this paper, we present a two-step method that classifies segments of the sensor data by their axis and change in values represented by peaks or valleys. The goal is to classify the swimming activities not by their time window but by the performance of the particular swimming activity's motions. Based on our findings, this approach reduces the issues associated with fixed window segmentation, and its flexibility allows our system to better support inexperienced swimmers as opposed to many of the tools available today. Further expanding on this research we can evaluate the performance of each activity individually and provide proper feedback.

II. Background

A. Limitations to Existing Swimming Activity Recognition

Research papers and commercial devices have focused on supporting professional and elite swimmers, generally individuals who are not beginners. Table 1 shows how the data are collected and how machine learning algorithms are developed/validated. Often, researchers get their participants for the user studies from national and college swim teams.

Larry Powell is with Texas A&M University, USA (e-mail: lpowellcs@gmail.com).

	Sensor Placement	Sensors	Application Populatoin	Data Segmentation	Study Types	Classification Method	Validation
Commercial Devices (Fitbit, Garmin, Apple Watch) [1,12,13]	Wrist	Gyro, Accelerometer	Hobbyist/Professional	Proprietary			
Qin et al.[7]	Wrist	Gyro, Accelerometer,	Elite/Professional 17	2 minutes time window	Controlled Study	Feature Extraction / Support Vector Machine	Cross Validation
Santos et al.[8]	Lower Back	Gyro, Accelerometer, Barometer, Heart rate, Magnetometer	Elite/Professional 10	1 second time window	Controlled Study	Feature Extraction / Machine Learning Algorithm	Cross Validation
Wang et al[9]	Lower Back	Gyro, Accelerometer, Barometer, Heart rate, Magnetometer	Elite/Professional 15	2 minute time window into 10 second windows Segment	Controlled Study	Feature Extraction / Hidden Markov Model (HMM)	Cross Validation
This paper	Lower Back	Acclerometer	Beginners - 15 Beginner - 5	6 seconds window into Peak/Valley Segment	Controlled Study Realistic Study	Feature Extraction / Machine Learning Algorithm	Leave-one-subject-out Validation

Fig. 1 Research and commercial representation for the data collection and machine learning algorithm development/validation in swimming activity recognition. The activities classified for all papers is backstroke, breaststroke, freestyle, and butterfly

Additionally, existing researchers mostly utilize time windows and segmentation methods, such as Support Vector Machine or Random Forest. We present a method that uses a time window as the initial segmentation, and from there, we re-segment the data based on the overall change of sensor data within the window.

Existing swimming activity recognition methods will often misclassify backstroke and freestyle with either the breaststroke or the butterfly because of the small differences in the swimming motions. This is a consequence in the limited data collection methods used to develop the HAR models. Researchers will collect data in a controlled setting where participant performs one activity at a time with an observer to mark the exact time when the activity is performed. Most of the participants in the study are either elite or experienced swimmers [7], [8]. It is easier to collect swimming data through experienced swimmers and swim teams that can provide consistent activities. However, the accuracy of classifying swimming strokes decreases as more beginner and intermediate swimmers use the system. Quin et al. presented a study that used elite swimmers using a wrist wearable device to get 99% accuracy [9]. Quin et al. only collected elite swimmers swimming each stroke in a collected setting. For our study, the control always performed better because the fatigue and exhaustion have not caused the swimmer's performance to decrease during each swimming stroke. They also mention in the paper that they will expect and get less accuracy when working with intermediate and beginner swimmers. We collected beginner swimmers once in a controlled setting and another in a freeform setting and got equally accurate results with additional swimming strokes like sidestroke and treading water.

B. Wrist-worn Commercial Devices Lack HAR Robustness

Commercial devices that use swimming activity recognition have been tailored to support regimented, lap swimming environments. Such features as stroke count, lap count, and lap time are standard among commercial devices like the Apple Watch and Fitbit are not that useful for novice swimmers. Lee et al. discovered in a user study to examine commercial devices' performance that the features have a 20% misclassification rate [10]. Because these commercial devices are produced by corporations, it is hard to understand the classification method used; however, for Fitbit, we discovered they use a four second time window for segmentation based on the patent [11]. We also discovered that the algorithms developed could only recognize these activities with high accuracy when used by experienced swimmers. Morais et al. analyzed and found that the Garmin watch recognizes all activities except butterfly at 92% accuracy [1]. Another researcher found that the commercial device's classification will reduce accuracy and performance if used by beginner and intermediate swimmers.

C. Peaks and Valley Swimming Analysis Provide Additional Information

Researchers discovered that peaks and valleys are associated with the swimming stroke's single motion. Bachlin et al. found that the peaks on the x-axis are specifically associated with the single motion of freestyle [12]. Fantozzi et al. further focused on using more specific phases in swimming and discovered that breathing patterns are found within the peaks and valleys based on sensor data [13]. Ohgi et al. found that when the swimmer gets more fatigued, the coordination of the individual motions decreases [14]. The individual's physical limb coordination is a shared feature when distinguishing between beginner and competitive swimming skill levels [14]. Based on all these previous papers, the importance of peaks and valleys can tell the difference between fatigue, breathing patterns, and proficiency level. For this reason, we built our system around peaks and valleys to support the classification of multiple activities in a shared window, which will enable advanced analytics of this nature.

D. Segmentation of Data by Time Window

Using a time window for activity recognition is the most common form of separating the data for the machine learning algorithms. However, some papers have split from a single time window to a multi-layer window method.

Qin et al. present a multi-window input method for their swimming activity classier [9]. The first window segments the overall data into large two-minute windows. The system then uses smaller ten-second window segments as the primary input for the machine learning algorithm. The second, smaller window is still relatively large at ten seconds, which is expected to have a high accuracy since the swimming stroke is performed multiple times within that window. We use a 6-second window capable of classifying at similar accuracy compared to overall large 2-minute window of Quin et al.

Ma et al. present a more adaptive time window in which the method expands and contracts the window based on if there is a change in the probability density of the data. The system's large overhead relies on constantly changing the window and will increase as more activities are incorporated [15].

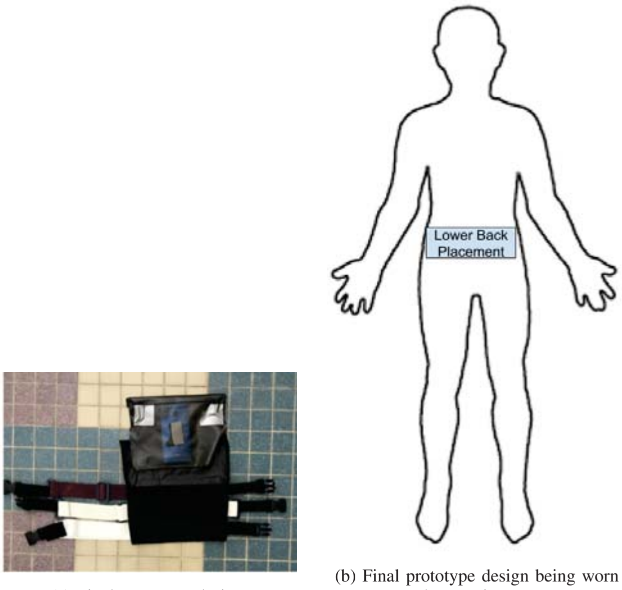
Cherian et al. present a two-window segment method where the first window gets the data and a second smaller window is used in the classification [16]. The system then selects the most frequent label from the smaller segmented windows as the output for the initial window.

Our work expands on previous works in this area that modify the window concept; however, instead of using smaller windows, we segment the data based on a median threshold which collects all values associated with peaks and valleys. This method supports a more adaptive approach while removing the overhead associated with constant window changes.

III. Methodology

A. System Implementation

The system used an HTC6500LVW mobile device that collected 3D accelerometer data locally with a BMA250 3-axis accelerometer that produced a range of values between -18 and 18 at 100Hz. The mobile device was stored in a waterproof pack and placed between the user's lower back as presented in Fig. 2. An android application was developed specifically for the studies and installed on the mobile device. The application collected the sensor data and stored the information on local storage within the device. The user could input the swimming stroke they performed, and the labeled data would be stored with the sensors data, providing observational data for the researcher to use.



(a) Final prototype design

(b) Final prototype design being worn by a swimmer

Fig. 2 Wearable device and placement location on the participant

B. Activities Captured

Our studies included lap swimming strokes (backstroke, breaststroke, butterfly freestyle). However, as we collected more data and read more papers on water safety, we included two other forms of propulsion-based swimming: treading-water and sidestroke [17], [18]. Treading water is a commonly required swimming activity that allows the person to be perpendicular to the water. Treading water is currently the only way for the swimmer to get their bearings while in a stationary position which is important for survival swimming. Sidestroke, however, is more common in survival swimming and is a swimming stroke that uses the least amount of energy. The way sidestrokes work is that the individual is on their side and uses an arm motion constantly underwater as propulsion.

C. Data Collection

For this work, the primary source of collecting participants was a university-level beginner's swimming course. New participants were recruited to participate in the study across two phases; all sessions shared the same system and sensor placement. All data were collected at the university pool, first in a controlled setting and second in a more freeform setting where they wore the device during class or individual practice.

1) Controlled Study: For the controlled study, we recruited a total of 15 participants from the *university's beginner swimming classes*. Each participant was requested to swim (backstroke, breaststroke, butterfly, freestyle, sidestroke, and tread water) for at most 50 meters or 2 minutes. We collected a total of 2.6 hours of swimming data and for each swimming stroke individual was: freestyle (32.86), backstroke (19.71), breaststroke (25.37), butterfly (35.65), sidestroke (19.56), and treading water (21.75).

2) Freeform Study: For the freeform environment, we recruited a total of five participants from the *university's*

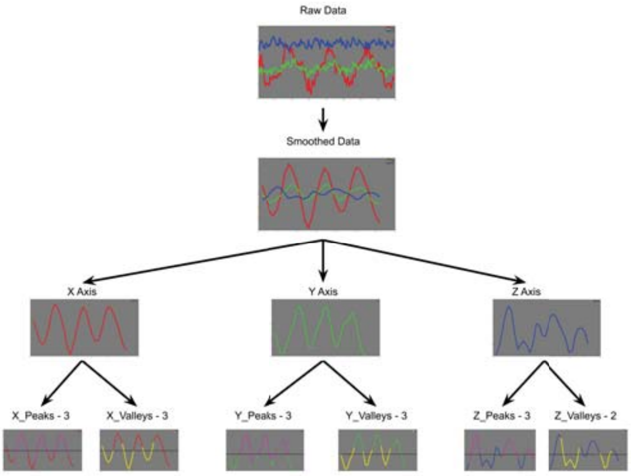


Fig. 3 Segmentation method of raw data to axis (X, Y, Z) specific peak/valley datasets: X-Axis (Red), Y-Axis (Green), Z-Axis (Blue), Peaks (Magenta), Valleys (Yellow)

beginner swimming classes. Each participant was requested to wear the wearable device during class or provide up to 30 minutes of swimming data. On the side, while the participant was swimming, a researcher observer would track the swimming stroke and the start and stop time. We collected a total of 2.06 hours of swimming data.

D. Activity Recognition Evaluation Method

For evaluating the performance of the machine learning model in human activity recognition, the following metrics are usually performed: accuracy, f1-score, precision, and recall [19]. Precision, recall, and f1-score are the most common when the data is imbalanced. With our data showing imbalanced data for each swimming stroke, we utilized the f1-score as the primary means of evaluating our swimming activity recognition system.

Once we found the optimal performance of the machine learning algorithms, we used the complete data set from user study 1 to build the machine learning models for the realist data from user study 2. For study 2, we analyzed all the possible groupings of machine learning models to determine the best axis and peak/valley that will produce the best overall results for classifying the swimming strokes. Because we evaluated each axis (X, Y, Z) and the peaks/valleys segments for each axis, we produced six machine learning models.

IV. Model Design

A. Six-Second Time Window

The time window is essential to the overall performance of the machine learning algorithms. Previous papers have presented that if the time window is too small or large, it can affect the accuracy of the classifier [6]. Using a time window is a suitable method and should not be removed when it comes to activity recognition. The reason is that the data need to be segmented to build a real-time classifier. For our system, the perfect window is not essential, and it does not have to be exact and must fit the activities being recognized. However, it does have to be large enough to contain the entire activity. We choose a six-second time window because on average a person performs a full swimming motion every three seconds and we wanted the window to be large enough to encompass multiple swimming motions within the segmented data.

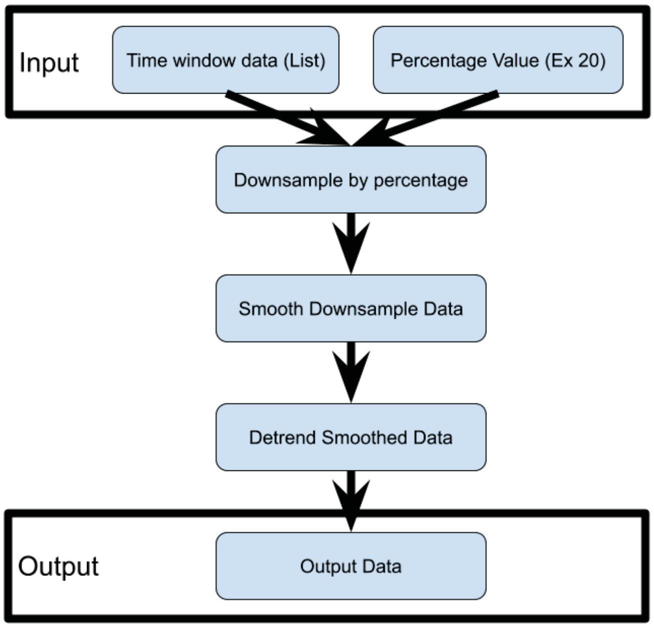


Fig. 4 Algorithm process for smoothing the data

B. Resampling and Smoothing

During prepossessing the data, it is not uncommon to smooth it before using it for activity recognition. The goal of smoothing data is to help reduce the effects of noise on the activity recognition model. We present a method for smoothing the data, which can be used for any time window length. Our algorithm uses Matlab2020B and its (*downsample*, *smoothdata*, and *detrend*) functions. An illustration of the algorithmic steps is presented in Fig. 4.

Our system begins with the raw data matching the frequency of the sensor collection tool. From there, we down-sample the data based on a frequency percentage; for example (20% of 100Hz is 5 points per second). We then smooth the data over that 1-second window to reduce the noise further. The smooth function we use is a Gaussian filter instead of a normal rolling mean filter. Finally, the data are sent to a *detrend* function, which removes the polynomial trend of the overall data so we can get the peak and valley data sets. We developed a dynamic smoothing algorithm to produce fine or coarse data from the time window.

C. Axis Peak/Valley Detection

When it comes to peak and valley segmentation, we build upon an already working algorithm. Sezgin et al. used a peak valley detection method for sketch recognition which calculates the peaks/valleys based on the points passing the median [20]. Our system builds on this and keeps all starting

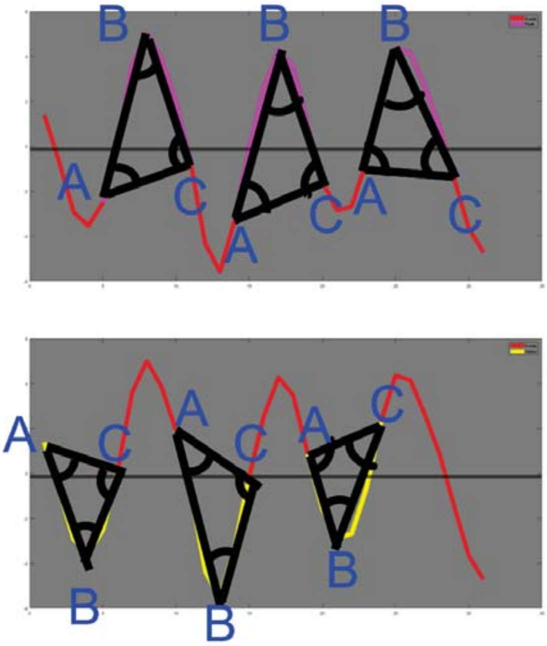


Fig. 5 Visual representation of peak/valley angles

points that passes the median as it finds the peak and gets the final point as it again crosses the median. The aforementioned Fig. 3 visualizes how each axis collects peaks and valleys segments.

D. Axis Feature Extraction

TABLE I
FEATURES FROM SKETCH RECOGNITION RESEARCH

#	Feature Name
(A)	Total Duration
(B)	Max Speed
(C)	Max Speed Squared
(D)	Min Speed
(E)	Min Speed Squared
(F)	Average Speed
(G)	Average Speed Squared
(H)	Total Intersection
(I)	End Intersection
(J)	Difference Intersection
(K)	Total Length
(L)	Bounding Box
(M)	Height Bounding Box
(N)	Depth Bounding Box
(O)	Volume Bounding Box
(P)	Diagonal Bounding Box
(Q)	Distance First to Last
(R)	Distance Total Length/Bouding Box Diagonal
(S)	Convex Hull
(T)	Sum Total Angle 3D Points
(U)	Sum Abs Total Angle 3D Points
(V)	Sum Squared Total Angle 3D Points
(W)	Speed Local Maxima
(X)	Speed Local Minimum
(Y)	Cosine Angel Between Points 1 and 3
(Z)	Sin Angel Between Points 1 and 3
(AA)	Cosine Angel Between Points First and Last
(BB)	Sin Angel Between Points First and Last

We use 55 features collected from sketch recognition research [21], [20], [22] as presented in Table I. We include an additional set of features from the activity recognition and signal processing domains, which are given in Table II. Most features in Table II are from the signal processing module that can be installed so that functions can be accessible in Matlab2020B software.

TABLE II
ACTIVITY RECOGNITION AND SIGNAL PROCESSING FEATURES

#	Feature Name
(A)	Entropy
(B)	Change In Direction
(C)	Spectral Kurtosis
(D)	Max Change in Mean
(E)	Min Change in Mean
(F)	Spectral Mean
(G)	Root Mean Square
(H)	Root Sum of Squares
(I)	Power bandwidth
(J)	Mean
(K)	Standard Deviation
(L)	Kurtosis
(M)	Change in Entropy
(N)	Direction Change Ratio
(O)	Peak/Valley Angle A
(P)	Peak/Valley Angle B
(Q)	Peak/Valley Angle C
(R)	Variance
(S)	Correlation Coefficient
(T)	Covariance

We include a feature that is angle based, utilizing the angles at the peak/valley to the start and endpoints as represented in Fig. 5. The resulting triangle illustrate how our algorithm selects the point on the starting or ending side of crossing the median, not an interpolated point on the media. Each angle is its own feature, and all features are generated individually for each axis (X, Y, Z) of the sensor, producing 285 features.

The least amount of features need to be extracted to build an optimal performing machine learning algorithm. We use a filtering method to remove all the features that are redundant. We use the Minimum Redundancy Maximum Relevancy (MRMR) algorithm. The *MRMR* function built with Matlab2020B provides a ranked value for each feature. We use a threshold of 0.001 and keep all the features that provide a relevance value above the threshold.

E. Machine Learning Model Development

We used the Matlab2020B *fitcauto* function to determine the optimal machine learning algorithm, which automatically selects the best models to train based on the feature data provided. For all six datasets, *fitcauto* selected ensemble, nearest neighbor (KNN), naive bayes (NB), support vector machine (SVM), regression tree, and decision tree. It created 150 variations of each using different sets of parameters, selecting the model with the lowest validation loss and highest accuracy.

We used leave-one-subject-out validation for evaluating the machine learning algorithm from the user study 1 datasets. Leave-one-subject-out validation is when we evaluate each machine learning model by using the entire dataset from a single participant as test data while the other participants are

TABLE III
TABLE OF SELECTED FEATURES FOR EACH AXIS

Axis	Peak/Valley	Feature Selected Count
X	Peak	7
	Valley	7
Y	Peak	7
	Valley	154
Z	Peak	9
	Valley	262

training data. The *fitcauto* function is used to build each model for the leave-one-subject-out validation. For each participant, we used *fitcauto* for each of the six grouped datasets (X-axis Peak, X-axis Valley, Y-axis Peak, Y-axis Valley, Z-axis Peak, Z-axis Valley). The entire dataset was used to produce the machine learning models to validate the performance of user study 2 datasets.

F. Segmentation Classification

For this paper, we present the results from the optimal selected models. As in previous literature, we use the majority-selected classifier from across the six machine learning models. Thus, from all the peak/valley segments for each axis within the time window, the most frequent label that was classified will be the final label. Fig. 6 is a representation of how the majority selection is performed in a time window.

V. RESULTS

A. Features Selected

The feature selection process used the entire dataset from user study 1. *MRMR* was used to rank all 385 features and provide a relevance Score. The *MRMR* function produces a predictor Score, which signifies the feature's importance to classification. We use a threshold of 0.001, and if any Score is above that value, it will be a kept feature. Table III shows the number of features selected for each axis and peak/valley. Among them, the valleys for both the Z and Y axis had a large number of features that were above 0.001.

B. Controlled Study Analysis

The overall goal of analyzing the controlled study was to determine if the peak/valley recognition algorithm produced an optimal performance. The way we did this is using the leave-one-out validation. Leave-one-out validation is where we use each participant individually as the test set, and all other participants are the training data. We collected 15 participants producing 15 separate F-Scores for each participant. We averaged all the F-Scores to get the final result for each axis and peak/valley group.

Using leave-one-out validation produced a high F-Score for each axis. For X, Y, and Z peak: **0.98**, **0.98**, and **0.86**, respectively. For X, Y, and Z valley: **0.98**, **1.00**, and **1.00**, respectively. We note the majority of results produced an F-Score of above 0.95, with only the exception of the Z-axis peak.

TABLE IV
SELECTED GROUPINGS OF AXIS AND PEAKS/VALLEYS MODELS

Grouped Data	Selected Datasets	F-Score
1	Valley Z Axis	0.95
2	Peak Z Axis Valley Z Axis	0.93
3	Peak Z Axis Valley Z Axis Valley Y Axis	0.96
4	Peak Z Axis Valley Z Axis Valley Y Axis Peak Y Axis	0.95
5	Peak Z Axis Valley Z Axis Valley Y Axis Peak Y Axis Valley X Axis	0.93
6	Peak Z Axis Valley Z Axis Valley Y Axis Peak Y Axis Valley X Axis Peak X Axis	0.92

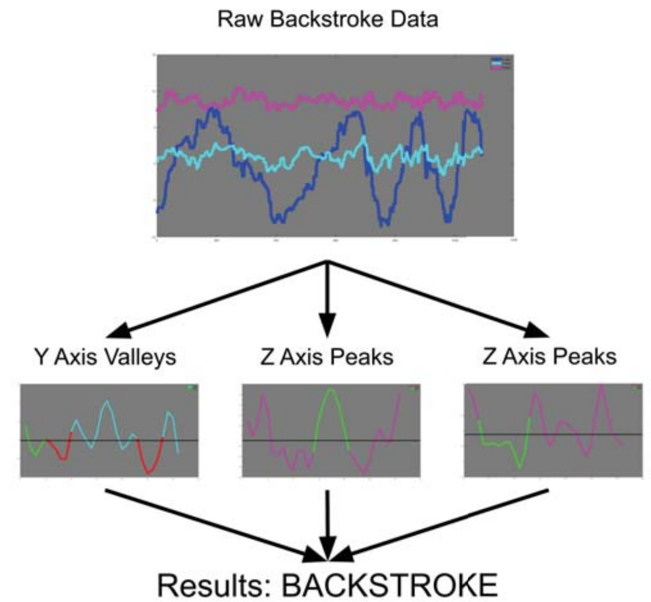


Fig. 6 Majority selection classification process of a real-time window (x-Axis (*Blue*), Y-Axis (*Cyan*), Z-Axis (*Magenta*)); for the peaks and valleys, the Correct Classification (*Green*) and Miss Classification (*Red*) are presented

C. Freeform Study Analysis

The controlled study results produced high F-Scores for all groupings of datasets. We wanted to determine which groups should be used and considered for a freeform study. The way we did this was to incorporate the majority classifier for the time window. We then examined all possible combinations of groups from a single group to using all six groups. We reviewed 63 total possible combinations.

We found that the groups built upon themselves and only incorporated another axis and peak/valley. The Z axis produces the best results for our dataset and grows to Y and X at the end, as presented in Table IV. The final best result is to use

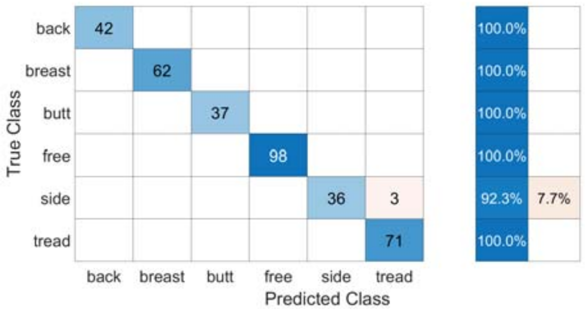


Fig. 7 Freeform study confusion matrix results using Z-Axis Peaks, Z-Axis Valleys, Y-Axis Valleys

TABLE V
COMPARISON OF TIME WINDOW *f-score* CLASSIFIERS TO THE SEGMENTED CLASSIFIER DEVELOPED FOR SWIMMING ACTIVITIES:

	Ba	Br	Bu	Fr
Garmin [7]	.98	.86	.94	.98
Wang et al. [23]	1.0	.92	.92	1.0
Time Window Only	1.0	1.0	.85	1.0
Peak/Valley Segmented Time Window	1.0	**1.0**	**1.0**	1.0

Br-breaststroke, Ba-backstroke, Bu-butterfly, Fr-freestyle

the Z-axis Peak/Valley and the Y-Axis Valley, which produced a 0.96 F-Score. The confusion matrix in Fig. 7 shows that the only issue presented is butterfly and breaststroke.

We compared our peak/valley segmentation method among other activity recognition paper results. We discovered that the butterfly is the most difficult activity to recognize based on Table V. Our system combats the issue and is able to classify beginner swimmers at near-perfect accuracy.

VI. Discussion and Future Work

When reviewing the sensor data's X, Y, and Z Axes, we discovered that the Z and Y axes were selected for providing the highest F-Score. The Z and Y axes had the most factors for breaststroke, butterfly, sidestroke, and treading water. Among the Z and Y axes, the valley classifiers produced the highest F-Score among all groupings. For some of the swimming strokes like breaststroke and butterfly, the peaks are points in which the person breaches the water while valleys are while they stay underwater. The swimmer has the most control when it comes to interactions underwater, while breaching relies on having to counter gravity as well as the waves [24]. Our results reinforce that for algorithms that use sensors to examine the swimmer's performance the motions that result in valleys within the datasets should be weighted more on the overall machine learning algorithm when classifying swimming activities.

Beginner swimmers lack coordination and proper form while they swim compared to more professional swimmers [25], which is a good reason why the butterfly is the hardest swimming stroke because it requires a lot of energy and coordination to perform [26]. Classifiers have issues with classifying butterflies because the swimmers often perform it wrong, causing the motion to look identical to breaststroke. Our system focuses on that issue and classifies based on all the strokes performed within the window. This will mitigate the algorithm's classification performance issues due to errors in swimmer's technique.

We plan to expand on the work and move from just the classification of repetitive activities to classifying single-motion activities. Another goal is to incorporate proficiency and error detection for each swimming stroke to help coaches and swimmers produce optimal physical coordination.

VII. Conclusion

The purpose of this paper was to use a two-form segmentation system that will remove the need for producing an optimal time window for swimming recognition. We did this by segmenting a time window dataset based on the groups of data that pass the median threshold. Results show that the segmentation method works and gives high F-Scores for all axes (X, Y, Z). We plan to move forward with developing a combination of the single and repetitive action classification system for swimming activity.

References

[1] J. E. Morais, J. P. Oliveira, T. Sampaio, and T. M. Barbosa, "Wearables in swimming for real-time feedback: A systematic review," *Sensors*, vol. 22, no. 10, p. 3677, 2022.

[2] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*. IEEE, 2006, pp. 4–pp.

[3] Q. Sun, E. Gonzalez, and B. Abadines, "A wearable sensor based hand movement rehabilitation and evaluation system," in *2017 Eleventh International Conference on Sensing Technology (ICST)*. IEEE, 2017, pp. 1–4.

[4] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Computing*, vol. 19, no. 5, pp. 26–35, 2015.

[5] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–27, 2010.

[6] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[7] R. Mooney, L. R. Quinlan, G. Corley, A. Godfrey, C. Osborough, and G. ÓLaighin, "Evaluation of the finis swimsense® and the garmin swim™ activity monitors for swimming performance and stroke kinematics analysis," *PloS one*, vol. 12, no. 2, p. e0170902, 2017.

[8] Z. Zhang, D. Xu, Z. Zhou, J. Mai, Z. He, and Q. Wang, "Imu-based underwater sensing system for swimming stroke classification and motion analysis," in *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE, 2017, pp. 268–272.

[9] X. Qin, Y. Song, G. Zhang, F. Guo, and W. Zhu, "Quantifying swimming activities using accelerometer signal processing and machine learning: a pilot study," *Biomedical Signal Processing and Control*, vol. 71, p. 103136, 2022.

[10] M. Lee, H. Lee, and S. Park, "Accuracy of swimming wearable watches for estimating energy expenditure." *International Journal of Applied Sports Sciences*, vol. 30, no. 1, 2018.

[11] Fitbit. (2022) Us-10918907-b2 - automatic detection and quantification of swimming. [Online]. Available: https://portal.unifiedpatents.com/patents/patent/US-10918907-B2

[12] M. Bächlin and G. Tröster, "Pervasive computing in swimming: A model describing acceleration data of body worn sensors in crawl swimming," in *2009 Joint Conferences on Pervasive Computing (JCPC)*, 2009, pp. 293–298.

[13] S. Fantozzi, V. Coloretti, M. F. Piacentini, C. Quagliarotti, S. Bartolomei, G. Gatta, and M. Cortesi, "Integrated timing of stroking, breathing, and kicking in front-crawl swimming: A novel stroke-by-stroke approach using wearable inertial sensors," *Sensors*, vol. 22, no. 4, p. 1419, 2022.

[14] Y. Ohgi, "Microcomputer-based acceleration sensor device for sports biomechanics-stroke evaluation by using swimmer's wrist acceleration," in *SENSORS, 2002 IEEE*, vol. 1. IEEE, 2002, pp. 699–704.

[15] C. Ma, W. Li, J. Cao, J. Du, Q. Li, and R. Gravina, "Adaptive sliding window based activity recognition for assisted livings," *Information Fusion*, vol. 53, pp. 55–65, 2020.

[16] J. Cherian, V. Rajanna, D. Goldberg, and T. Hammond, "Did you remember to brush?: a noninvasive wearable approach to recognizing brushing teeth for elderly care," in *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*. Barcelona, Spain: ACM, 2017, pp. 48–57.

[17] B. A. Morrongiello, M. Sandomierski, D. C. Schwebel, and B. Hagel, "Are parents just treading water? the impact of participation in swim lessons on parents' judgments of children's drowning risk, swimming ability, and supervision needs," *Accident Analysis & Prevention*, vol. 50, pp. 1169–1175, 2013.

[18] R. K. Stallman, K. Moran Dr, L. Quan, and S. Langendorfer, "From swimming skill to water competence: Towards a more inclusive drowning prevention future," *International Journal of Aquatic Research and Education*, vol. 10, no. 2, p. 3, 2017.

[19] W. Sousa Lima, E. Souto, K. El-Khatib, R. Jalali, and J. Gama, "Human activity recognition using inertial sensors in a smartphone: An overview," *Sensors*, vol. 19, no. 14, p. 3213, 2019.

[20] T. M. Sezgin, T. Stahovich, and R. Davis, "Sketch based interfaces: Early processing for sketch understanding," in *ACM SIGGRAPH 2007 Courses*, ser. SIGGRAPH '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 37–es. [Online]. Available: https://doi.org/10.1145/1281500.1281548

[21] B. Paulson and T. Hammond, "A system for recognizing and beautifying low-level sketch shapes using ndde and dcr," in *ACM Symposium on User Interface Software and Technology (UIST2007)*, 2007.

[22] R. Blagojevic, S. H.-H. Chang, and B. Plimmer, "The power of automatic feature selection: Rubine on steroids." *SBIM*, vol. 10, pp. 79–86, 2010.

[23] Z. Wang, X. Shi, J. Wang, F. Gao, J. Li, M. Guo, H. Zhao, and S. Qiu, "Swimming motion analysis and posture recognition based on wearable inertial sensors," in *2019 IEEE international conference on systems, man and cybernetics (SMC)*. IEEE, 2019, pp. 3371–3376.

[24] B. Burkett, R. Mellifont, and B. Mason, "The influence of swimming start components for selected olympic and paralympic swimmers," *Journal of applied biomechanics*, vol. 26, no. 2, pp. 134–141, 2010.

[25] Y. Matsuda, Y. Yamada, Y. Ikuta, T. Nomura, and S. Oda, "Intracyclic velocity variation and arm coordination for different skilled swimmers in the front crawl," *Journal of human kinetics*, vol. 44, p. 67, 2014.

[26] M. Hołub, A. Stanula, J. Baron, W. Głyk, T. Rosemann, and B. Knechtle, "Predicting breaststroke and butterfly stroke results in swimming based on olympics history," *International journal of environmental research and public health*, vol. 18, no. 12, p. 6621, 2021.