Optimization of deep neural network-based human activity recognition for a wearable device

K. Suwannarat, W. Kurdthongmee *

School of Engineering and Technology, Walailak University 222 Thaibury, Thasala, Nakornsithammarat, 80160, Thailand

ARTICLE INFO

Keywords:
Human activity recognition
Deep neural network
Wearable device

ABSTRACT

Human activity recognition (HAR) attempts to classify performed activities from data retrieved from different sensors attached to the body. Most publications pertaining to HAR based on deep neural networks (DNNs) report the development of a suitable architecture to improve recognition accuracy by increasing the parameters of the architecture. Our work follows a different approach by attempting to optimise DNN-based HAR by reducing the dimensions of acceleration data, by finding a suitable sample size for processing by the DNN and by reducing the parameters of the proposed architecture. The experiments rely on employing two previously presented DNN-based HAR architectures as the baselines and starting points to create our candidate architectures. The variations in the dimensions of acceleration data, i.e., \{xy, yz, xz, x, y, z\}, and the sample size, i.e. \{4, 6, 8\} s duration, to these candidate architectures are experimented to produce the winner architecture which takes the shortest sample size and the minimal dimensions of acceleration data while preserving the recognition precision. The results indicate that despite the number of parameters is approximately half of the baseline architecture with two dimensions of acceleration data and shorter sample size (i.e., using a sample of 4 s duration instead of 8 s and only the \(xy\) axes of acceleration data), the resulting DNN-based HAR classifiers can produce comparable or better recognition precision than the baseline classifiers. The experimental results were obtained using three different popular datasets: the WISDM, the UCI HAR, and the Real World 2016. The proposed classifiers with optimised settings are useful as they require less processing time and reduce power consumption, both in terms of retrieving acceleration data from the sensor and the CPU processing time. Furthermore, they reduce the memory requirements for parameter storing and are suitable for incorporation in a wearable device.

1. Introduction

Human activity recognition (HAR) is a challenging problem for the research community. The ultimate aim of HAR is to monitor the elderly and serve healthcare purposes [1] because HAR is a form of assistive technology. The applicability of HAR could be expanded markedly by using it in combination with other technologies such as wearable devices and the Internet of Things. HAR requires the use of sensors, smartphones, or images [2,3]. The data collected from these sensors are collectively or separately used to classify the activities into the classes of walking, running, sitting, sleeping, standing, and abnormal activities. These activities can be further used for medical analysis, keeping track of elderly people, monitoring crime, military actions, and smart home applications. Currently, because of the popularity of wearable devices, data are always retrieved from these devices in the form of acceleration. Additionally, because the features of sensors can be enhanced by integrating a gyroscope, a wearable device can also process the orientation and angular velocity of the device and, in turn, its wearer. The sophistication of wearable devices and a smart algorithm to process their data makes it possible to accurately perform HAR. However, this is offset by the high demand for battery power. Additionally, in certain cases, the algorithm requires additional processing power. This prevents or inhibits the execution of other tasks running on the wearable device.

Deep neural networks (DNNs) have been proposed for object classification and detection within an input image. This is a problem of two-dimensional processing, and DNNs have successfully served to solve it. A DNN is useful because it does not require knowledge to tailor features to be classified and detected. This means that an expert feature engineer is not required. Owing to their popularity and success, DNNs have been applied to the one-dimensional classification and detection domain, which includes HAR. DNN based HAR is superior to ordinary pattern recognition algorithms because it can also detect high-level or context-
aware activities such as having coffee [4], in addition to simple activities such as walking or running.

Most publications on the research topic of DNN-based HAR have proposed different DNN architectures, that is, different numbers of layers, kernel sizes, numbers of filters, and type of operators, which give rise to a variation in the total number of parameters with the aim of improving the overall recognition performance measures of all activities [5, 6, 7] and eliminating the restrictions on the sensor attachment position [8]. As a starting point, our research relies on two different DNN architectures, those of Paulo [9] and Brownlee [10], to create classifiers with acceptable recognition precision. These architectures are useful for our study because their source codes, wholly in Python, are provided. This makes it easier to follow them in order to modify some parts to be suitable for benchmarking or extending. The DNN architectures are used as baselines to conduct the experiments in which the number of axes of the acceleration parameters is varied. This is achieved by selecting only a single axis or a pair of axes of acceleration data and varying the sample sizes that affect the size of an input feature vector of the DNN. Additionally, the baseline architectures are modified to decrease the total number of parameters while preserving the recognition precision. Our intention is to optimise these parameters to make it possible to implement the classifier on a mobile device with moderate performance. Once it is implemented and deployed, the DNN-based HAR application would have to consume the least possible power, although the recognition precision would be compromised. With respect to studying the power consumption of mobile devices, an Android smartphone was selected. Specifically, Khan et al. [11] reported that 16.9% on average is required if all axes of data from both the accelerometer and gyroscope sensors are used together for a HAR application. Liu et al. [12] studied the application of economical microelectromechanical systems (MEMS) inertial measurement units (IMUs), which integrate an accelerometer and a gyroscope within the same package, to a wearable running device. They found that the power consumption of the IMU MEMS gyroscopes had active currents in the milliampere range. The power consumption of the MEMS IMU sensors is a function of the sampling rate. Furthermore, the active current of an accelerometer may increase by over an order of magnitude at high sampling rates, which is required to obtain high HAR accuracies.

We hypothesise that using the minimum number of axes of acceleration data in addition to a small sample size with a less complex architecture can result in recognition precision comparable to that achieved using the baseline architecture with acceleration data of all axes and a long sample size. Reducing the number of dimensions of the acceleration data would be beneficial as it would shorten the size of the feature vector during activity recognition. This is because the size of the feature vector is the product of the sample size S and the axes of acceleration data A, that is, it is equal to $80 \times 3$ for a sample size of 80 when all axes of acceleration data, $x$, $y$, and $z$, are used. Several commercially available IMU MEMS, for example, ADXL362 [13], can provide a low-power mode in the microampere range during their operation. Further reducing the total number of parameters of the architecture reduces the total memory requirement, which is beneficial for a wearable device with restricted memory size. These findings could further reduce the power consumption of wearable devices and appropriately enhance the functionality of these devices so that they are multifunctional.

The remainder of this paper is organised as follows. In Section 2, the materials and methods are detailed. This is followed by the experimental results and discussion in Section 3. Finally, the paper is concluded in Section 4.

2. Materials and methods

Before providing the details in this section, the following terms are defined: architecture and classifier. Architecture refers to the DNN architecture, which consists of different layers, such as dense or convolutional layers. The classifier is a DNN inferencing program that uses the model that is the result of the training of the DNN architecture.

2.1. Dataset

Most publications in the field of HAR have attempted to create a classifier with high recognition precision. For training and validating a classifier, they make use of three publicly available datasets: the WISDM Smartphone and Smartwatch Activity and Biometrics dataset (WISDM) [14], Human Activity Recognition Using Smartphones dataset (UCI HAR) [15], and the Real World 2016 from the University of Mannheim dataset (Real World 2016) [16]. To ensure that our hypothesis is correct and works across different settings of sensor attachment and experimental setups, all of these datasets were used in the course of our experiments. The WISDM dataset consists of labelled accelerometer data from 29 participants captured while they were performing daily activities. The captured activities are as follows: walking, jogging, ascending stairs, descending stairs, sitting, and standing. The WISDM lab researchers then aggregated the time-series data into samples that summarise the user activity in intervals of 10 s. All accelerometer data were retrieved from an embedded tri-axial accelerometer within a mobile device running an Android operating system. Participants carried the device in the front left pocket of their trousers. During data capturing, the accelerometer was sampled every 50 ms (20 samples per second). The WISDM dataset contains 1,098,207 samples covering six activities with the following distribution of samples: walking: 424,400 (38.6%), jogging: 342,177 (31.2%), ascending stairs: 122,869 (11.2%), descending stairs: 100,427 (9.1%), sitting: 59,939 (5.5%), and standing: 48,395 (4.4%).

The UCI HAR dataset was collected from a group of 30 volunteers aged 19–48 years. Each person was requested to perform six activities: walking, ascending stairs, descending stairs, sitting, standing, and laying while wearing a Samsung Galaxy S II smartphone on their waist. The smartphone had an embedded accelerometer and gyroscope; 3-axial linear acceleration and 3-axial angular velocity were sampled at a constant rate of 50 Hz. The experiments were video-recorded and the videos were used to annotate the data manually. The obtained dataset was then randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data. It can be observed that the last activity differs between the WISDM and UCI HAR datasets. This means that the architecture needs to be retrained specifically with each dataset to create a proper classifier.

The Real World 2016 dataset captured acceleration, GPS, gyroscope, light, magnetic field, and sound level data of the following activities: walking, ascending stairs, descending stairs, sitting, standing, laying, jumping, and running in real-world settings. These activities are different from the captured activities of the previous two datasets. There were fifteen subjects, eight males and seven females, with average age, height, and weight of $31.9 \pm 12.4$, $173.1 \pm 6.9$, and $74.1 \pm 13.8$, respectively. For each activity, the researcher simultaneously recorded the acceleration of the body positions of the chest, forearm, head, shin, thigh, upper arm, and waist. Each subject performed each activity roughly 10 min except for jumping, which took about 1.7 min due to the physical exertion. The data are equally distributed between males and females. Each movement was recorded by a video camera to facilitate usage. It is noted that we only make use of the acceleration from the sensors attached to the waist in our experiments. The obtained dataset was randomly partitioned into two sets, where 70% was selected for training and 30% for validation.

These datasets were pre-processed in preparation for training the architectures, the baselines and our proposed ones, to produce the models to be used by the classifiers. The pre-processing details are as follows: (1) acceleration data normalisation, (2) segment creation from the acceleration data (sample size defines the number of segments per feature vector), and (3) training and validation datasets splitting with a ratio of 70:30.
2.2. Method

The experiment was designed to study the recognition performance measures in terms of precision, recall, and F1-score of all classifiers resulting from training the architectures with the three datasets detailed in the previous section by using a single axis, \( x \), \( y \), or \( z \), and pairs of axes, \( xy, xz, \) or \( yz \), of acceleration data. The results are compared to those obtained using all \((x, y, z)\) axes of acceleration data. Because we attempted to minimise the number of training parameters, the orientation and angular velocity data from all datasets were not considered in the experiments. The experiments were conducted by varying the sample size, which is defined as the number of acceleration data values required to create a segment that is used as an input to the architecture during training and validation and deploying the classifiers. The following sample sizes were used in the experiments: 40, 80, 120, and 160. The sample size affects the performance of the classifiers because larger sample sizes require more time for capturing and processing, with high CPU utilisation and power consumption for both CPU and acceleration sensors. The training parameters, which are called a weight file in some publications, were also minimised by reducing the number of features and kernel sizes of the architectures. A weight file has a high impact on the memory usage of the classifier during its deployment. If it can be reduced, it means that the classifier can be implemented on a hardware platform with high memory restrictions. Although, we attempted to minimise several parameters, the recognition performance measures were required to be comparable to the ones of the state-of-the-art baseline classifiers. Two main online tutorials on the topic of DNN-based HAR are presented by Paulo [9] and Brownlee [10]. Let us name them Paulo and Brownlee architectures. Both architectures were employed as the baselines in our research because all implementation details, explanations, and their fully functional source codes were available. The total number of layers of both architectures in these tutorials is equal, but they make use of different sets of operators and arrangements. The Brownlee architecture differs from the Paulo architecture, as it was originally designed to be trained with all six dimensions of acceleration data (three dimensions each of total acceleration and body acceleration) along with all three dimensions of the gyroscope data. Additionally, the Brownlee architecture relies on using datasets from the University of California, Irvine (UCI) machine learning repository [17]. The Brownlee classifier was reported to have an average recognition precision of 91% and the Paulo classifier of 95%. The details of these two architectures are given briefly in comparison with our proposed ones, which produce promising recognition performance measures.

Most publications proposing a new architecture do so without giving details of the underlying design approach to their new architecture in terms of the number of operators and their arrangement, the details on the number of parameters, and the setup of the training hyperparameters. It is commonly mentioned that the new architecture was designed based on a trial-and-error approach. In our research, we relied heavily on the Paulo and Brownlee architectures. It is arguable that while there have been many publications in the field of HAR, why our research was based on benchmarking with online tutorial materials. From our review, we found that Kwapisz et. al [14], provided a comparative recognition accuracy of different approaches applied to HAR using the cellular phone accelerometer problem. Fortunately, some of the approaches are based on using the WISDM dataset. It is stated in [18] that a one-to-one comparison with existing approaches for HAR is not feasible because they use different devices, datasets, and activities, and new implementation is inevitably required. However, we only refer to the comparative recognition precision detailed in Table 1 as a guideline to support using the Paulo and Brownlee architectures as our baselines. This comes from the fact that both architectures provide comparable or better recognition precisions than the previously presented approaches. Additionally, after we followed their tutorials, we found that both classifiers produced an impressive set of recognition accuracies. Lastly, our intention was not to propose an architecture with higher recognition performance measures. However, we intended to propose one with lower complexity in terms of parameters, operators, and axes of acceleration data, that relies on using shorter sample size.

As seen in Table 2, both the Paulo [9] and Brownlee [10] architectures make use of multiple convolutional layers with dropout and pooling layers. Both architectures have an equal number of layers. The main difference between them is the number of features and kernel sizes of the convolutional layers. In terms of the number of parameters, the Paulo architecture has a higher number of parameters, 344,646. This is more than twice that of the Brownlee architecture. To fulfil our requirement to lower the total number of parameters, during the course of our experiments, several classifiers were designed in such a way that their number of features in each layer is half or less than half of the Paulo architecture.

Moreover, the kernel sizes were varied and assigned in the following ways: (1) fully inheriting them from the Paulo architecture, (2) keeping them constant in all layers, (3) rearranging them in increasing order, and (4) rearranging them in decreasing order. For example, one of our classifiers has the following architecture: Conv (80, 12), Conv (64, 10), Conv (48, 8), Conv (32, 6), MaxPooling (2), DropOut (0.5), and Dense (Softmax). Only the activation function of the regularised linear unit is used in all convolution layers of our proposed architectures. These architectures were then trained with different settings of axes of acceleration data and validated with respect to all three datasets. The recognition performance measures were recorded and compared with the Paulo and Brownlee classifiers, which were used as the baselines.

The details of the most common experimental settings for each of the architectures are as follows:

- the axes of acceleration data: \((xyz, xy, xz, yz, x, y, z)\),
- the number of iterative computations for each setting: 10,
- the number of epochs 1500 with the ratio between the training and test datasets: 67:33,
- the initial random state (of sklearn’s train test split function): 42,
- the sample sizes: \((80, 120, 160)\) which are equivalent to \((4, 6, 8)\) s with half the sample size overlapping the sample window,
- the loss function and optimiser: (categorical cross entropy, rmsprop)

Because there were many architectures to be trained apart from both of the baselines, to accelerate the training stage, the experiments were conducted on the Google Cloud Platform on the following hardware: Intel
Broadwell CPU with a single NVIDIA Tesla K80 with the main usage packages: Python’s sklearn and Tensorflow with Keras (with GPU support). Upon performing preliminary experiments with double iterative computations and epochs of 150, several settings of our proposed architectures were removed from further consideration. Only four promising architectures, our proposed #1 - #4, were kept and used in our experiments. These are summarised along with the two baseline architectures in Table 2 and illustrated their relative architectural differences in Figure 1.

Followings are the step-by-step details of our experimentation applied to each dataset:

1. We trained the two baseline and four proposed architectures with the experimental settings detailed earlier. The total number of experiments for the four proposed architectures were 4 \times 6 \times 3 \times 10 = 720 (4 architectures, 6 combinations of axes of acceleration data, 3 sample sizes, 10 iterative computations), and for the two baseline architectures were 2 \times 10 = 20 (2 architectures trained with all axes of acceleration data, sample size of 160, and 10 iterative computations). It should be noted that the baseline architectures were trained with a fixed sample size of 160 with all axes of acceleration data in order (1) to avoid modification to their original architectures and (2) to conform to the settings originally provided by their designers.

2. We computed the average precisions by activities of each architectures from all 10 iterative computations. For a dataset with six activities, this created the following results:

Table 2. Architectures of DNN-based HAR presented in [9,10] and our proposed architectures. The parameters in parentheses next to the convolutional layers are the number of features and the kernel size. All the convolutional and dense layers rely on using the RELU and softmax, respectively, as their activation functions.

| Layer | Architectures | Paulo | Brownlee | Our proposed #1 | #2 | #3 | #4 |
|-------|---------------|-------|-----------|-----------------|----|----|----|
| 1     | Conv (160, 12) | Conv (64, 3) | Conv (128, 12) | Conv (64, 10) | Conv (64, 12) | Conv (64, 24) | Conv (96, 6) |
| 2     | Conv (128, 10) | Conv (64, 3) | Conv (96, 10) | Conv (64, 10) | Conv (96, 18) | Conv (64, 12) | Conv (32, 10) |
| 3     | Conv (96, 8) | Dropout (0.5) | Conv (64, 8) | Conv (32, 8) | Conv (64, 12) | Conv (64, 12) | Conv (32, 10) |
| 4     | Conv (64, 6) | MaxPool (2) | Conv (32, 6) | Conv (14, 6) | Conv (32, 6) | Conv (14, 12) | Conv (32, 6) |
| 5     | MaxPool (2) | Flatten | MaxPool (2) | MaxPool (2) | MaxPool (2) | MaxPool (2) | MaxPool (2) |
| 6     | Dropout (0.5) | Dense (100) | dropout (0.5) | Dropout (0.5) | Dropout (0.5) | Dropout (0.5) | Dropout (0.5) |
| 7     | Dense | Dense | Dense | Dense | Dense | Dense | Dense |

Parameters (all) | 346,566 | 128,898 | 189,446 | 84,246 | 151,814 | 77,032 |
(dual axes) | 187,910 | 83,112 | 150,534 | 76,456 |
(singale axes) | 186,374 | 81,960 | 149,254 | 75,880 |

Figure 1. The relative architectures of DNN-based HAR presented in [9,10] and our proposed architectures.
3. Results and discussion

In this section, the experimental results are presented and discussed with respect to each dataset used for training and validating the classifiers. Before presenting the experimental details, let us define a short notation to refer to a classifier. The notation is in the following format:

- **P_{a,s,d}** represents the average precision by activity from 10 iterative computations by the activities of the baseline architectures.

- **B_{a}** is the minimum of average precisions across all activities of all baseline classifiers.

- **P_{a,s}** is the minimum of average precisions across all activities of all our proposed classifiers.

- **P_{a,s,d}** is the minimum of average precisions across all activities of all our proposed classifiers.

3.1. Experimental settings and results

For ease of understanding, the approach to find the winner classifier detailed above is explained step by step for the case of the winner classifier with the WISDM dataset and a sample size of 120. Figure 2(a) illustrates the first two steps of the approach. All graphs are plotted from the experimental settings with an equal sample size of 120. Each graph shows the results for each combination of axes of the acceleration data. Within the graph, the bar graphs with the same colour are the average precisions by activities from 10 iterative computations of a classifier. The light orange and light purple represent the average precisions from 10 iterative computations by the activities of the Paulo and Brownlee classifiers, respectively. At this point, let us focus only on the line graphs. First of all, we create **B_{b}** by computing the average precisions across all activities of the two baseline classifiers, i.e., the averages of the light orange and light purple line graphs. Then, the minimum value between these two average precisions, **B_{m}**, is determined. **B_{m}** is represented as a horizontal black line in the figure. Now, considering all bar graphs, each set of the same colour bar graphs that belong to each classifier is visited in order to calculate a single average precision across all activities: **P_{a,s,d}**. The results from this step are shown in Figure 2(b). The same colour is used to relate the line graph and bar graphs to the classifier. For example, the purple line graph is a single average precision across all activities of the proposed classifier #1 whose average precision by activity is illustrated by purple bars. Finally, it can be concluded that our proposed classifier #1 produces two candidate classifiers, which are the ones trained with **xy** and **xz** axes of the acceleration data. This comes from the fact that its average precision across all activities, i.e. the purple horizontal line, is above the **B_m** line in the **xy** and **xz** graphs in Figure 2(b). To determine the final winner, it is necessary to consider the recall and/or F1-score of these two winner candidates.

The next section presents the results of the experiments with the above-mentioned architectures with all datasets and all the settings.

4. Conclusion

In conclusion, we have presented a novel approach to find the winner classifier by considering the average precision (y-axis) for the WISDM dataset trained with double axes of acceleration data with sample size of 120. For each proposed classifier, the same colour bar graphs represent the average precision by activity. For the baseline classifiers, the line graphs represent their average precisions by activities. The black horizontal line is the minimum of average precisions across all activities of the baseline classifiers. For each proposed classifier, its average precision across all activities is represented by the horizontal line with the same colour as the bar graph of the corresponding classifier.
(classifier name, axes of acceleration data, sample size). For our proposed classifier, we refer to each of them by using only an index #n. For example, (#1, x, y, z) is our proposed classifier #1 trained with x, y, z axes of acceleration data with a sample size of 160.

First of all, since the aim of our research is to find the most appropriate classifier for a wearable device, one factor which is critical for such a device is the computational performance of the classifier. This factor is independent of the usage dataset. It depends on the operations within the classifier, the sample size, and the number of axes of acceleration data. We performed experiments to measure the computational performance of both the baseline classifiers and all these combinations of our proposed ones. Some results are presented in Figure 3 in the form of relative computational performances to the Brownlee architecture whose computational performance is the best. Only the classifiers which will be referred later on in this section are presented in the figure. It can be seen that all relative computational performances of our proposed classifiers are in between the Paulo and Brownlee architectures. One classifier, which is (#2, 2-axis, 120), produces better performance compared to the Brownlee architectures.

For the WISDM dataset, the experimental results are detailed in Table 3. It was found that our proposed classifier #1 produced two winner candidates with two experimental settings: (#1, x, y, z) and (#1, x, y, z, 160). The results indicate that although the first and second candidate classifiers were trained with a sample size of 120 and 160 and with the x, y axes of acceleration data, they performed better than both the baseline classifiers in 3 out of 6 activities: sitting, standing and lying. However, they could only produce comparable results to both baseline classifiers for the activities of upstairs and downstairs. Both candidate classifiers failed to outperform both baseline classifiers for the activity of walking. On average, the candidate classifiers gave rise to better average classification result. It may be argued that the improvement in terms of the average precision is very low, at only 1%. However, if the parameters of the architectures are considered, our proposed classifier #1 uses a lower number of parameters compared to the Paulo classifier. Additionally, although the Brownlee classifier whose number of parameters is slightly lower than and produces superior computational performance to our proposed classifiers #1, it requires processing all axes of acceleration data. These altogether make it more promising. From these two candidate classifiers, we performed a performance comparison using recall and F1-score, which determined a single winner, (#1, x, y, z). The comparison results are illustrated in Figure 4. From the figure, it is observable that most of the performance measures of the (#1, x, y, z) classifier are comparable to or better than both baseline classifiers with the exception of the activities of walking and downstairs.

The experimental results of the baseline and our candidate classifiers for the UCI HAR dataset are shown in Table 4. Our set of winner candidate classifiers is: {(#1, x, y, z), (#3, x, y, z), (#1, x, y, z, 160)}. It can be seen that the average precisions across all activities of our candidate classifiers are between the Paulo and Brownlee classifiers, and their average precisions are all equal, although our candidate classifiers failed to produce comparable results to both baseline ones for the activities of walking and upstairs. A performance comparison using recall and F1-score was then required. Finally, this produced a single winner, (#1, x, y, z). The comparison results are illustrated in Figure 5. It becomes clear that the majority of performance measures of our winner classifier are between both baseline classifiers. If the variances of the results, shown as the vertical lines at the top of the bar graph of each activity and their averages, are considered, the performance of the winner classifier is comparable to that of the Paulo classifier and is clearly superior to that of the Brownlee classifier. For the UCI HAR, it can be summarised that our winner classifier is useful only from the point of view of its lower total number of parameters within the architecture, as it makes use of only x, y axes of acceleration data. Its performance measures are between the Paulo and Brownlee classifiers.

Finally, with respect to the Real World 2016 dataset, several candidate classifiers were produced. These are shown in Table 5. It can be seen that all winner candidate classifiers used only a sample size of 120. Surprisingly, even the candidate classifier (#1, z), which made use of a single axis of acceleration data, is capable of producing equal average precision to the baseline classifiers in a majority of activities. That is to say, it lacked behind the baseline classifiers for the activities of jumping and climbing up. However, the winner classifier is the (#1, x, y, z). Its average precision is mostly comparable or better than the Paulo and Brownlee classifiers with the exception of the activities of climbing up and sitting. The comparison results illustrated in Figure 6 confirm that all performance measures agree with the precision.

At this point, it can be summarised that when all datasets are considered, the shared winner classifier is our proposed classifier #1, which was trained with x, y axes of acceleration data with sample sizes of either 120 or 160. It not only produces mostly superior performance measures to the baseline architectures but also has an equal total number of layers and has a total number of parameters between the baseline architectures. It dominates from the point of view that it processes only the x, y axes of the acceleration data which significantly reduces its relative computational performance to be between the two baseline classifiers. Additionally, the experimental results from these three different datasets confirm that the proposed architecture and its resulting classifier are independent of the type and model of the acceleration sensors. However, our experiments only covered the datasets with data from acceleration sensors attached to the waist of the subjects. For other attachment locations, more experiments are required to be conducted with the previously detailed approach. Additionally, the experimental

![Figure 3](image-url) The relative computational performances between the Paulo and some of our proposed architectures to the Brownlee architecture.
settings can also be extended to cover higher numbers of epochs, different sample sizes, loss functions, and optimisers. These are open questions for future research.

4. Conclusion

In this study, it is hypothesised that DNN-based HAR can be optimised to rely on using fewer axes of acceleration data with a short sample size and fewer total parameters while preserving the activity recognition precision. This would open the opportunity to implement the resulting classifier on a wearable device with restricted resources. Additionally, a reduction in the size of the feature vector, which is the product of the sample size and the number of axes of acceleration data, is expected to produce a classifier that is computationally more efficient and consumes less battery power. Two previously presented DNN-based HAR architectures with shallow layers were selected and used as baseline architectures in experiments to test the hypothesis. Several architectures with equal number of layers but fewer parameters to these baseline

| Activities    | Average precision | Baseline classifiers | Our proposed #1 with xy dimensions |
|---------------|-------------------|----------------------|-----------------------------------|
|               |                   | PP                   | BL                               | Sample size: 120 | Sample size: 160 |
| Walking       | 96                | 99                   | 90                               | 91                |
| Upstairs      | 99                | 96                   | 98                               | 99                |
| Downstairs    | 97                | 99                   | 97                               | 97                |
| Sitting       | 88                | 90                   | 94                               | 93                |
| Standing      | 97                | 97                   | 98                               | 98                |
| Lying         | 89                | 85                   | 92                               | 91                |
| Average       | 94                | 94                   | 95                               | 95                |

Figure 4. Comparison of the average performance measures between the baseline and winner (#1, xy, 120) classifiers across activities with the WISDM dataset.
architectures were designed. These DNNs were trained with several settings with variations in the axes of the acceleration data \{x, y, z, xy, xz, yz\} and sample sizes (80, 120, and 160) from three popular HAR datasets, namely, WISDM, UCI HAR, and Real World 2016. For each dataset, the average precision across all activities calculated from ten iterative computations per setting was used to select a single winner classifier. The results support the hypothesis as they clearly indicate that even the classifier that was constructed with fewer parameters and trained with two axes of acceleration data and a shorter sample size achieves comparable to better recognition performance measures than the baseline.

Table 4. Our winner classifiers trained with acceleration data from the UCI HAR dataset.

| Activities | Average precision |
|------------|-------------------|
| Baseline classifiers | Trained with xy dimensions |
| PP | BL | Sample size 120 | Sample size 160 |
| | | Our proposed #1 | Our proposed #3 | Our proposed #1 |
| Walking | 98 | 94 | 79 | 79 | 80 |
| Upstairs | 98 | 99 | 89 | 90 | 91 |
| Downstairs | 80 | 76 | 81 | 80 | 81 |
| Sitting | 80 | 77 | 93 | 93 | 93 |
| Standing | 95 | 90 | 99 | 98 | 99 |
| Lying | 94 | 85 | 97 | 98 | 96 |
| Average | 91 | 87 | 90 | 90 | 90 |

Figure 5. Comparison of the average performance measures between the baseline and winner \(#1, xy, 120\) classifiers across activities with the UCI HAR dataset.
The common winner classifier with respect to all datasets has approximately half of the total parameters of the baseline one. Its convolutional layers consist of fewer features and smaller kernel sizes, which is ideal for smartphone deployment. The presented findings can also be used as a guideline to select the appropriate classifier and experimental settings for a specific wearable device. In addition to proposing a new DNN architecture, this paper suggests an alternative approach to optimise or improve performance measures. The details for searching the

| Activities       | Baseline precision | Sample size 120 |
|------------------|--------------------|-----------------|
|                  | FP | BL | #1  xy | #1  xx | #2  xx | #3  xx | #1  yy | #1  x | #1  z |
| Climbing up      | 62 | 69 | 60   | 61    | 86    | 87    | 58    | 75    | 61    |
| Climbing down    | 82 | 90 | 80   | 91    | 72    | 71    | 79    | 83    | 83    |
| Jumping          | 98 | 79 | 98   | 98    | 93    | 94    | 72    | 97    | 77    |
| Laying           | 94 | 87 | 96   | 99    | 98    | 98    | 98    | 100   | 100   |
| Running          | 84 | 74 | 89   | 89    | 83    | 79    | 91    | 96    | 95    |
| Sitting          | 81 | 81 | 65   | 77    | 74    | 75    | 85    | 56    | 86    |
| Standing         | 53 | 53 | 52   | 56    | 67    | 66    | 56    | 51    | 57    |
| Walking          | 88 | 82 | 84   | 82    | 68    | 72    | 87    | 76    | 86    |
| Average          | 80 | 77 | 78   | 82    | 80    | 80    | 78    | 79    | 81    |

**Figure 6.** Comparison of the average performance measures between the baseline and winner (#1, xy, 120) classifiers across activities with the Real World 2016 dataset.
winner classifier from different experimental settings by comparing the performance measures with the baseline classifiers are also presented.

Declarations

Author contribution statement

W. Kurdthongmee: Conceived and designed the experiments; Performed the experiments; Wrote the paper.
K. Suwannarat: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Funding statement

This work was supported by the Office of the National Digital Economy and Society Commission (grant no. BK1017/63).

Data availability statement

Data will be made available on request.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

[1] C. Jobanputra, J. Bavishi, N. Doshi, Human activity recognition: a survey, Proc. Comp. Sci. 155 (2019) 698–703, the 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019). The 14th International Conference on Future Networks and Communications (FNC-14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 14)/Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050919310166.
[2] B. Ni, Y. Pei, P. Moulin, S. Yan, Multilevel depth and image fusion for human activity detection, IEEE Trans. Cybern. 43 (5) (2013) 1383–1394.
[3] A. Bayat, M. Pomplun, D.A. Tran, A study on human activity recognition using accelerometer data from smartphones, Proc. Comp. Sci. 34 (2014) 450–457, the 9th International Conference on Future Networks and Communications (FNC-14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 14)/Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914008643.
[4] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey, Pattern Recogn. Lett. 119 (2019) 3–11, deep Learning for Pattern Recognition. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865518300045X.
[5] Y. Chen, Y. Xue, A deep learning approach to human activity recognition based on single accelerometer, IEEE Int. Conf. Syst. Man Cybern. (2015) 1488–1492, 2015.
[6] A. Ignatov, Real-time human activity recognition from accelerometer data using convolutional neural networks, Appl. Soft Comput. 62 (2018) 915–922 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917305665.
[7] N. Jaoseed, N. Boujnah, M.S. Bouhilel, A new hybrid deep learning model for human action recognition, J. King Saud. Univ. Comp. Inform. Sci. (2019) [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157819300412.
[8] B. Almaslukh, A.M. Artoli, J. Al-Muhtadi, A robust deep learning approach for position independent smartphone-based human activity recognition, Sensors (2018).
[9] P. Pinheiro, Human Activity Recognition - 1d Cnn, 2018. https://www.kaggle.com/paulopinheiro/humanactivity-recognition-1d-cnn/data/.
[10] J. Brownlee, How to Develop 1d Convolutional Neural Network Models for Human Activity Recognition, 2018. Sep. https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/.
[11] I. Khan, S. Khurtoo, S. Ali, J. Ahmad, Sensors are power hungry: an investigation of smartphone sensors impact on battery power from lifelogging perspective, Bahria Univ. J. ICT 9 (2016) 8–19, 12.
[12] Q. Liu, J. Williamson, W. Mohrman, K. Li, Q. Lv, R. Dick, J. Shang, Gazelle: energy-efficient wearable analysis for running, IEEE Trans. Mobile Comput. (2016) 10.
[13] A.D. Inc, Adxl362 Datasheet, 2019. https://www.analog.com/media/en/technical-documentation/datasheets/ADXL362.pdf.
[14] J.R. Kwapisz, G.M. Weiss, S.A. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explor. Newsl. 12 (2) (2011) 74–82 [Online]. Available:.
[15] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, in: In 21st European Symposium on Artificial Neural Networks, ESANN, 2013. Bruges, Belgium, April 24–26, 2013, 2013. [Online]. Available: http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2013-84.pdf.
[16] T. Snyder, Sensor-based Human Activity Recognition: Overcoming Issues in a Real World Setting, 2019. Ph.D. dissertation, Mannheim, Germany, http://ub-mannheim.bi b.uni-mannheim.de/49914/.
[17] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. Reyes-Ortiz, A Public Domain Dataset for Human Activity Recognition Using Smartphones, 2013.
[18] G. Bhat, N. Tran, H. Shill, U.Y. Ogras, w-har: An activity recognition dataset and framework using low-power wearable devices, Sensors 20 (18) (Sep 2020) 5356 [Online]. Available:.