Project Topic : Deep learning for physical activity recognition using wearable sensors

Submitted by: Team India
Date Submitted: 18 May 2022

This project aims to explore the effects of different preprocessing choices on the accuracy of three deep learning models, which are CNN, LSTM, and Bi-directional LSTM and on three publicly available human activity recognition datasets (UTD-MHAD, USC-HAD and UCI-HAD datasets). Three separate experiments were conducted to understand the effect of resampling techniques, window overlap rates, and image generation techniques.

In the first two experiments relating to resampling and window overlap rates, the Leave-One-Subject-Out Cross-Validation method was used. The preprocessed inputs were then passed to three baseline models (CNN, LSTM, and Bi-LSTM). For the third experiment concerning image generation techniques, training and test sets were split by subject based. The raw signal data were transformed to four images by employing Markov Transition Field, Gramian Angular Field, and Recurrence Plot, with the final image as a combination of all three image generation techniques. The generated images were then passed to the baseline CNN model as well as a hybrid CNN+LSTM model. For all experiments, the models were evaluated using accuracy as the metric.

Results show that resampling has a positive effect on all the deep learning models given a 128 resampling size. In terms of window overlap rates, the CNN model performs better without window overlap whilst the two LSTM models achieve better accuracy scores with a larger overlap rate. Lastly, image generation techniques are able to improve the performance of the baseline CNN model for the UTD-MHAD dataset but this is not the case for the USC-HAD and UCI-HAD datasets.

1. Introduction

The purpose of Human Activity Recognition (HAR) is to be able to understand the human action being performed from the acquired data automatically. HAR has many successful applications across different industries from healthcare, security, military, to personal health monitoring. Many people are familiar with this technology through the proliferation of wearable technologies with companies such as Fitbit, Apple Watch, and Garmin. These devices are able to monitor and track our heart rate, oxygen levels, and even sleep patterns. For specialised applications in healthcare, HAR has been used to support the elderly through fall monitoring systems and emergency alarms as well as provide a measure of rehabilitation progress amongst stroke patients (Chernbumroong et. al., 2013; Panwar et. al.,2019).

One of the earliest works in HAR goes back to 1999 and research into this field still remains strong today given the various applications mentioned above (Foerster, 1999). There have been many developments in the field with regards to the type of sensor used: video-based or sensor-based and the machine learning approaches applied (Wang et. al., 2018; Zhang et. al., 2022). Sensor-based HAR are preferred these days given its developments in sensor technology which improved its size and integration with the user as well as lowered cost and power consumption. Video-based sensors face issues relating to privacy, pervasiveness, and complexity. It is difficult to attach video-based devices to a subject whilst successfully capturing various daily activities and video processing techniques are computationally expensive (Lara and Labrador, 2013).

Traditional machine learning approaches used methods from the field of signal processing for feature engineering. However, these handcrafted features require domain expertise and a large amount of human effort which makes the approach too costly and unscalable. Deep learning (DL) approaches have since been applied to the HAR problem which advanced the field further. DL has the capability to automatically learn the most relevant features from the raw data with little domain knowledge. The power of DL also lies in its ability to approximate any function if it is given a reasonably large network and dataset. (Zhang et. al., 2022)

This project aims to explore three DL models, CNN, LSTM and Bi-directional LSTM, and investigate the impact of various preprocessing choices on the accuracy of these models on three publicly available HAR datasets - UTD-MHAD, USC- HAD, and UCI-HAD. This report is organised as follows: Section 2 provides the current state of the HAR problem where we can draw gaps from current studies and also provide a basis of our investigation. Section 3 presents our research question as drawn from the literature review. Section 4 illustrates the experimental design and methods of the project. Section 5 reports the experimental results. Section 6 discusses our results. Lastly, Section 7 provides recommendations for future work and summarises the findings of the project.

2. Literature Review

There are many works relating to the human activity recognition problem using wearable sensor data. It is challenging to evaluate the different deep learning approaches given the variability in the choice of data pre-processing, model and its parameters, and datasets to perform evaluation. For the purposes of this project, the literature review will allow us to identify gaps in the literature to base our research question on and provide us some benchmarks in evaluating the DL models we will develop.

2.1 Signal Images

Data for HAR is commonly collected from inertial sensors such as accelerometers and gyroscopes. An accelerometer measures the acceleration of a body in its own instantaneous rest frame. A gyroscope measures the orientation and angular velocity of a movement. For each sensor, data can be represented by a set of three vectors $v_i = (x_i,y_i,z_i)$, where $i = (1,2,3,...,n)$ for each time step. For HAR, one accelerometer or gyroscope reading is insufficient to be able to categorise the human activity in question (Gu et.al., 2021). Therefore we will look at current approaches on how to transform these raw signal data that will be applicable to DL models.

Jiang and Yin (2015) proposed to generate an activity image based on the combined raw signals of an accelerometer and a gyroscope. Data from the accelerometer is split further into two features - linear acceleration, which contains information on the body motion, and gravity. These 3 separate raw data are then stacked row-by-row to produce a signal image. This process is explained in detail in their paper under Algorithm 1. Finally, a 2D Discrete Fourier Transform is applied to the newly generated signal image where the magnitude is selected as the input to their proposed Deep Convolutional Neural Network (DCNN). Jordao et. al. (2019) however identified that their method is computationally expensive as the input size increases exponentially. In addition, Yang et. al. (2022) found that their approach is limited to the UCI dataset with 9 input signals, which is one of the datasets this project will explore. This method is therefore not suitable for our project as we aim to be able to apply the same transformation across all three datasets .

Jordao et. al. (2019) highlighted that the widely used Semi-Non-Overlapping-Window process to generate a set of data samples, also called as temporal windows, leads to biased results. This arises because there is a 50% overlap in the data between windows which are passed through cross validation. Consequently, 50% of the sample is found in both the training and test sets simultaneously. This ultimately results in a drop by around 10% in reported accuracy scores for models that employ this process. To resolve this bias, Jordao et. al. (2019) put forward two processes: Full-Non-Overlapping Window and Leave-One-Trial-Out. One obvious downside of the Full-Non-Overlapping Window approach is a decrease in the data sample generated given that there is no longer overlapping data. The Leave-One-Trial-Out is a more attractive approach when the dataset is small. In this approach, the Semi-Non-Overlapping-Window process is applied to generate the temporal windows and then a 10-fold cross validation was used to ensure that the same trial (also called as a single activity) does not exist in both training and test sets. In their experimentation to find out which of their proposed sample generation process achieves the highest accuracy score across existing CNN frameworks and datasets, the Leave-One-Trial-Out combined with cross validation achieved 84.66% when employing Chen and Xue (2015)'s model on the USCHAD dataset. None of the CNN frameworks evaluated were successfully implemented on the UTD dataset. This is one of the datasets we will experiment on.

Yang et. al. (2022) builds on the approach proposed by Jiang and Yin (2015) to be able to support any number of raw inputs. They proposed two methods: single column and multi-column activity graph. In the first method, after stacking the raw data row-by-row as performed by Jiang and Yin (2005), the last sequence is paired with the first sequence horizontally or in other words, the first sequence is stacked by row below the last sequence. This action is performed until all sequences are adjacent to other sequences at a minimum of one instance. As a result, the size of input is expanded height-wise. The second method has a similar underlying algorithm but instead of only generating data vertically, data is also expanded width-wise. The sorting algorithms are clearly published in their

paper under Algorithm 1 and 2 which allows for reproducibility. Their proposed multi-column activity graphs in combination with an optimised CNN model were able to achieve 90.17%, 87.70% and 64.12% on the UCI, USCHAD, and UTD datasets, respectively.

2.2 Image Generation

Aside from directly forming signal activity images, image transformation techniques can be employed for time series data. Amongst them are Gramian Angular Field (GAF), Markov Transition Field (MTF) and Recurrence Plots (RP). In GAF, the time series data is mapped to a polar coordinate system rather than the typical cartesian coordinate system. The transformation in GAF has the advantages of bijective transformation which has a one-to-one correspondence between elements and preservation of temporal relations. Due to these characteristics, the image formed through GAF can revert back to the original raw time series data (Ahmad and Khan, 2021). In MTF, the Markov transition probabilities are leveraged to sequentially preserve the temporal dimension of sensor signals. However, compared to GAF, MTF is not able to recover the original time series data due to its larger inverse image within the mapping function. They also have distinct capabilities such that GAF converts static information whilst MTF captures dynamic information (Wang and Oates, 2015). In RP, the recurrence structure of a time series data is captured. The data is converted into an image which represents the distances between each time step (Ahmad and Khan, 2021).

These techniques have been used widely on different time-series signals but research on its use within HAR still remains thin. Within the medical field, Zhang et. al. (2019) employed GAF to transform 1D ECG data to a 2D GAF image which is then passed to a PCAnet model to classify Myocardial Infarction. GAF and MTF techniques have also been applied in fault diagnosis from vibration signals in industrial machines (Han et. al., 2021). Amongst these three techniques when applied to various time series datasets from the UCR Archive , Debayle et. al. (2018) found that RP image transformation of time series data combined with a CNN model, outperformed GAF and MTF techniques also paired with a CNN model. Our project can add to this research by testing it on HAR datasets.

Within HAR, Xu et. al. (2020) criticised that other activity data preprocessing such as matrix rearrangement faces interpretability issues and Fourier transform requires higher computing resource which may not be ideal for the HAR problem. On the other hand GAF resolves these two issues such that it allows for greater interpretability and less computing time to apply. The most relevant work is done by Ahmad and Khan (2021) where they tested four different types of image generation techniques on inertial sensor data: Signal Images (SI), GAF, MTF and RP Images. They tested these techniques on a CNN model to extract features, specifically ResNet-18, and the learned features are fused using the canonical correlation fusion method. Lastly, the fused features are sent to a SVM classifier. Their model was applied on two datasets - UTD and HHAR. Our project can be compared with their results on the UTD dataset. For the UTD dataset, they found that GAF, MTF, and RP improved SI-based model accuracies by 0.5%, 0.2%, and 0.4%, respectively. Our project can add to the literature with regards to the effectiveness of image generation techniques on other datasets such as UCI HAR and USC HAD.

2.3 Convolutional Neural Network

Convolutional neural network (CNN) benefits from having the characteristics of local dependency and scale invariance when applied to a time series classification task (Wang et. al., 2019). Local dependency matters for the HAR task as neighbouring signal data has a high chance that

it correlates with the current signal data. Scale-invariance is also a useful trait to be able to detect the same movement but at a different frequency.

Jiang and Yin (2015) introduced a DCNN composed of two layers. The first layer has 5 kernels of size 5x5 with a 4x4 mean-pooling layer whilst the second layer has 10 kernels of size 5x5 with a 2x2 mean-pooling layer. Parameters such as kernel, bias, weight, and softmax coefficients were updated through stochastic gradient descent. In addition, SVM classifiers were trained to apply uncertainty in some activity classes. They employed accuracy and computational cost to evaluate their architecture. Their proposed architecture achieved accuracy scores between 95%-99% when applied to three public datasets (UCI, USC, and SHO).

Chen and Xue (2015) modified a conventional CNN architecture with 3 convolutional layers of 18, 36, and 24 filters and 3 max-pooling layers. In the first layer, the kernel size applied is 12x2 and 12x1 on the remaining two layers. Their model achieved an accuracy score of 93.8% on a large dataset from one accelerometer with 8 activities. Their model shows that a relatively simple model can also achieve good results.

Yang et. al. (2022) designed a CNN model and provided a detailed description of their parameter and hyperparameter choices in Table II of their paper. Similar to Jiang and Yin (2015), there are two layers with the first layer of 20 kernels of size 10x10 and the second layer of 30 kernels of size 7x7. The type of sub-sampling used was max-pooling. In determining the final size and quality of the activity image graphs, it was found that the two key parameters were aspect ratio and dots per inch. The optimal settings were 3:3 and 120 DPI, respectively. Their approach improved the average recognition accuracy by about 5% compared with other deep learning models. On the UTD dataset, it achieved up to 10% greater accuracy compared with other models. It would be interesting to compare our model results with their results from the UTD dataset.

2.4 Recurrent Neural Network and Long Short Term Memory

Long Short Term Memory (LSTM) is a type of recurrent neural network which is particularly useful in analysing time series data. As HAR is a time-series classification problem, RNN is able to capture temporal dependencies in the sensor data.

Tufek et. al. (2020) explored three types of LSTM models on the UCI dataset: 2 Layers, 3 Layers, and Bi-directional LSTM. They looked at testing different types of LSTM models due to the limitation of a standard LSTM network where the future input cannot be obtained from the current state. A bidirectional LSTM network overcomes this limitation as it connects two hidden layers of opposite directions to the same output (Tufek et. al., 2020). For the LSTM models, they consist of 32 neurons for each layer, use of 10-fold cross-validation, a 0.0025 learning rate, and a 0.0015 loss account coefficient, binary cross-entropy for optimization, and a dropout rate of 20%. The Bi-directional model has 2 layers with 100 and 32 hidden layers, respectively. Their experiment showed that the 3 Layer LSTM performed best at 97.4% accuracy on the UCI HAR dataset whilst the Bi-directional model achieved the lowest accuracy score of 90.3%.

Zhao et. al. (2018) developed a residual bidirectional LSTM model and tested it on the UCI HAR dataset. The use of residual networks allows gradients to pass through more layers directly and enables them to bypass layers. Hence, a deeper neural network can be developed. The authors emphasised that a deeper network requires regularisation to avoid overfitting whilst increasing accuracy. In comparing the accuracy of three LSTM models on the UCI HAR dataset, the bidirectional LSTM outperformed the baseline LSTM model by 0.3% and the residual bidirectional LSTM performed 2.8% better than the baseline model, achieving 93.6%.

2.5 Data Preprocessing

Aside from transforming the sensor data to images, there is a need to segment the raw signals into subgroups where each segment contains sufficient information to allow the DL model to classify the activity at any given time. Segmenting the raw signals can be done in two ways - non-overlapping windows or overlapping windows. When segments do not overlap, this means that information in one window is not contained in other windows. In overlapping windows, one window contains a percentage of information from the previous window. For instance, given a window of 128 readings (from 50 Hz sampling rate of a 2.56 second activity) a 50% overlap means that 64 readings from the previous window forms part of the data of the next window.

In the HAR literature reviewed above, there is minimal exploration of the effects of their pre-processing choices to the accuracy of their models. Yang et. al. (2022) is one of the few papers that discussed their results from varying sliding window overlap rates. The authors concluded that for smaller datasets such as the UTD dataset, a greater overlap rate is beneficial as it can generate more data samples. Consequently, the optimal overlap rate for the UTD dataset is 65% whilst the USC HAD and UCI datasets can achieve good accuracy results with a 50% overlap rate. In the Zhao et. al. (2018) LSTM models, they uncovered a surprising observation from their study. They found that window size is a key parameter in their models such that a smaller window size restricts the continuity of information and a large window size can cause classification errors. However, their paper did not include any rigorous testing on the optimal window sizes for their chosen dataset.

3. Research Question

The current literature on HAR focuses heavily on developing novel deep learning techniques whilst overlooking the effects of data preprocessing. This project aims to explore the effects of different preprocessing choices on the accuracy of three DL models - CNN, LSTM, and Bi-directional LSTM. The literature has shown the effectiveness of CNN in the HAR problem due to its local dependency and scale invariance. Two types of RNN models, LSTM and Bi-directional LSTM, were chosen for the networks' ability to capture temporal dependencies. Following the work of Tufek et. al. (2020) and given the limitation of a standard LSTM network, we also want to test our hypothesis on a bidirectional LSTM network.

In the first experiment, with regards to combining raw signals into one sequence for each observation, as explored by Jiang and Yin (2015) and Yang et. al. (2022), we will explore whether resampling has an effect on the accuracy of the three DL models. Resampling prevents data leakage from one sample to another and ensures equal lengths for each observation which will allow the DL models to classify better.

In the second experiment, we will investigate varying window overlap rates. Overlapping windows increases the number of data points available compared to non-overlapping windows, which can increase the predictive accuracy of DL models. This project will explore if these two hypotheses hold true across the DL models and HAR datasets. The three publicly available datasets that will be used are UTD-MHAD, USC-HAD, and UCI-HAD. However, the first two research aims can only be tested on two datasets - UTD-MHAD and USC-HAD given that the UCI-HAD dataset is a preprocessed dataset. More details will be provided in Section 4.

Another aim of this project is to explore the effects of different image generation techniques such as Gramian Angular Field, Markov Transition Field and Recurrence Plot. Given the limited

research on image generation techniques in HAR and their outperformance against signal images on two datasets (UTD-MHAD and HHAR) following the research of Ahmad and Khan (2021), this project intends to understand the effects of image generation on the accuracy of the CNN model. In addition to the CNN model, we will also test against a hybrid CNN-LSTM model as CNN alone will only capture the spatial features and not the temporal characteristics of the sensor signals. We hypothesise that combining all three image generation techniques and applied to the hybrid model will generate the best results because each image generation technique captures different information from the raw sensor signals.

Our research questions are as follows:

A. What is the effect of resampling sizes on the accuracy score of CNN and LSTM models?
B. How does varying window overlap rates affect the accuracy score of CNN and LSTM models?
C. What is the effect of applying image generation techniques on the accuracy score of CNN and CNN-LSTM hybrid models?

4. Data and Methods

4.1 Datasets

This project utilises three publicly available HAR datasets. The first one is named UTD-MHAD. It contains 4 data modalities: RGB videos, depth videos, skeleton joint positions, and inertial sensor signals. This project only utilises the inertial sensor data which includes both triaxial acceleration and gyroscope signals. The combined sensor data form a total of 6 features and were sampled at 50 Hz. In aggregate, there are 861 data sequences available. There are 27 distinct actions which are performed by 8 individuals, with each individual performing the same action 4 times. The actions covered range from basic arm movements such as swiping left/right to more scenario and sport specific movement such as basketball shooting. More details shown in Section 4.2. There are two different placements of the inertial sensor to collect the data. For action 1 to 21, the inertial sensor was placed on the individuals' right wrist whilst for actions 22 to 27, the sensor was placed on the individuals' right thigh. In order to control for the data collection process within this dataset, only activities 1 to 21 were chosen. (Chen et. al., 2015)

The second dataset is called USC-HAD. It contains 12 different activities which are defined as basic daily activities such as walking, running, and sitting. There are 14 individuals from which the data was collected and they performed each activity 5 times. In terms of data collection, one MotionNode was used and placed on the individuals' front right hip. A MotionNode is a 3 degrees of freedom inertial measurement unit which is typically used in 3D motion tracking (Motion Workshop Company, 2021). It has a sampling rate of 100 Hz. The data comes in the form of tri-axial accelerometer and gyroscope signal data which also forms 6 features. (Zhang and Sawchuk, 2012)

The third dataset was acquired from the UCI Machine Learning Repository. This dataset contains 6 different activities with the full list shown in Section 4.2. There are 30 individuals involved. Data collection was sourced from a smartphone which was attached to the individual's hip. Similar to the other two datasets, the data contains body acceleration and gyroscope signals. However, in contrast to the other two datasets, there is an additional feature which corresponds to total acceleration. Therefore there are 9 features in total. It has a sampling rate of 50 Hz. (Anguita et. al., 2013). The dataset is available as a preprocessed data with the following attributes: noise filtered, fixed-width sliding windows of 2.56 seconds and 50% overlap rate, and 70%/30% training-test split.

Below is a summary table of the key attributes of each dataset:

	UTD-MHAD	USC-HAD	UCI-HAD
Sampling Rate	50 Hz	100 Hz	50 Hz
Number of Individuals	8	14	30
Number of Activities	21	12	6

4.2 Exploratory Data Analysis

In this section, the aim is to summarise the main characteristics of the three datasets which may help us uncover interesting patterns as well as better understand the type of activities that need to be classified.

Figures 1 to 3 present the distribution of activity types across the three datasets. It shows that the UTD-MHAD dataset has a largely balanced distribution across all the activities. USC-HAD and UCI-HAD are imbalanced with USC-HAD displaying a larger degree of variability than the UCI-HAD dataset.

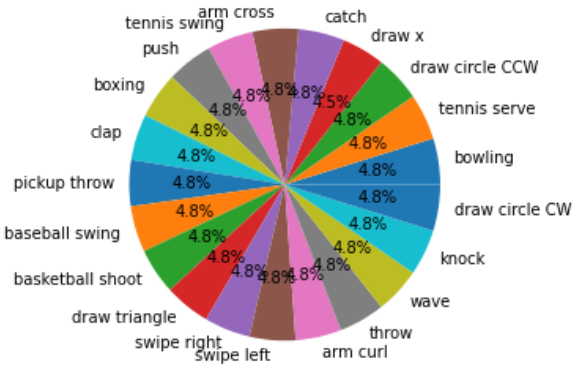


Figure 1. Pie chart of distribution by activity type for UTD-MHAD

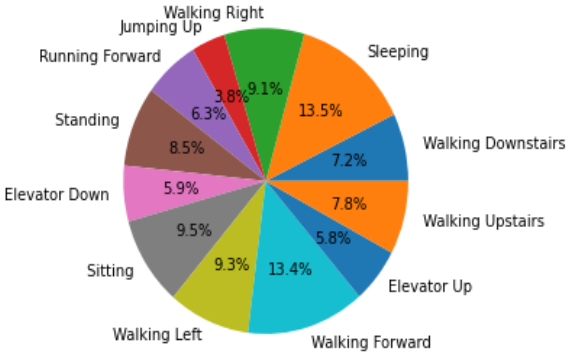


Figure 2. Pie chart of distribution by activity type for USC-HAD

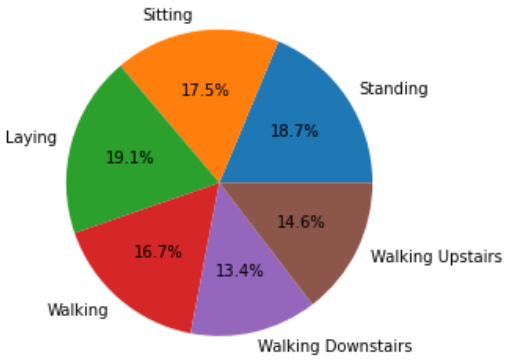


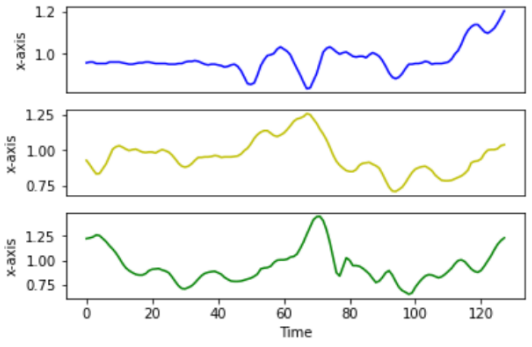
Figure 3. Pie chart of distribution by activity type for UCI-HAD

In terms of the types of activities presented, there are some similarities between the USC-HAD and UCI-HAD datasets such that they include basic daily activities such as walking, sitting, and standing. On the other hand, the activities covered in the UTD-MHAD dataset are more specific and complex. For instance, baseball swing and tennis swing are sport specific activities.

Plotting some samples of the raw signals from the accelerometer relating to the three axes can help visualise the variations in the signals across these activities. In particular, it is interesting to see whether there are more variations amongst the basic daily activities as found in the USC-HAD and UCI-HAD datasets or amongst the specific activities in UTD-MHAD.

Figure 4. Variation in x-axis accelerometer signals for USC-HAD for 3 subjects

a) Walking Forward



b) Sitting

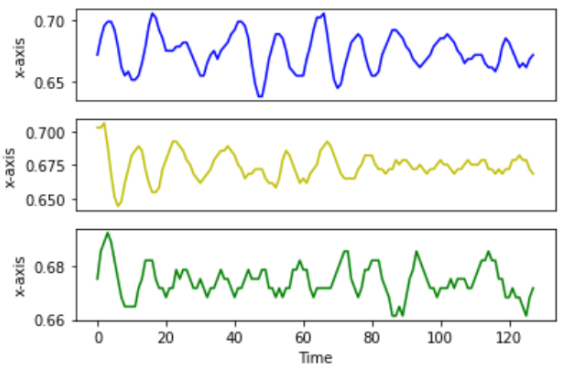
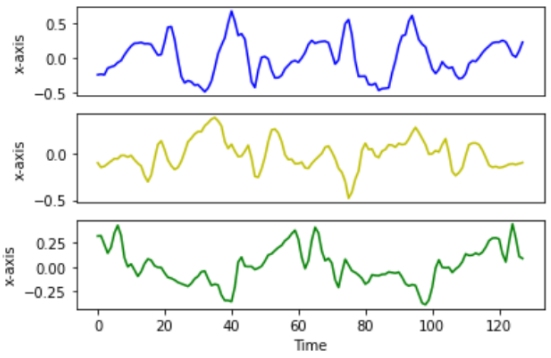
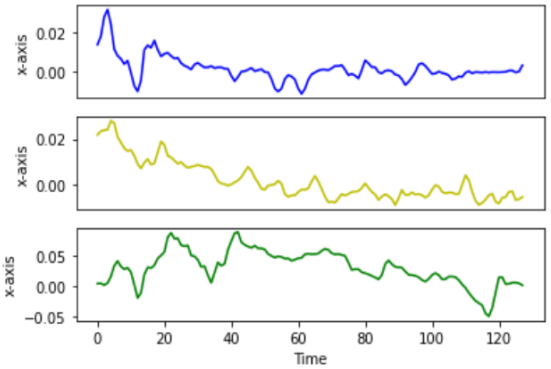


Figure 5. Variation in x-axis accelerometer signals for UCI-HAD for 3 subjects

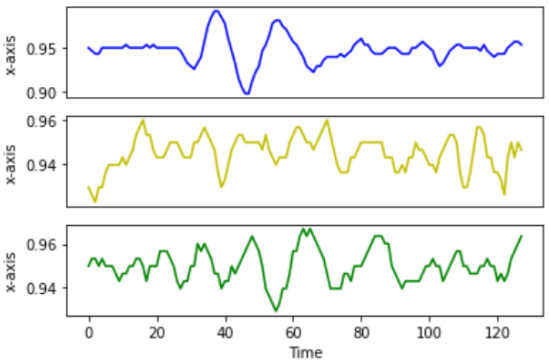
a) Walking



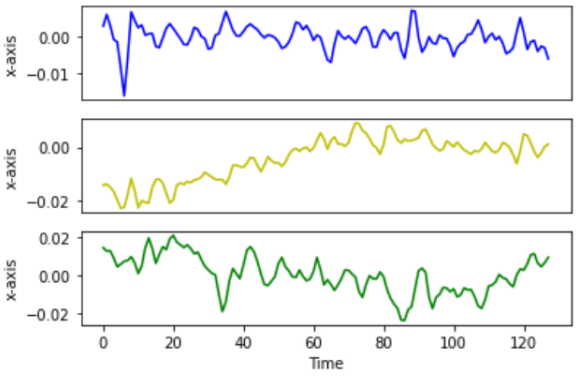
b) Sitting



c) Standing



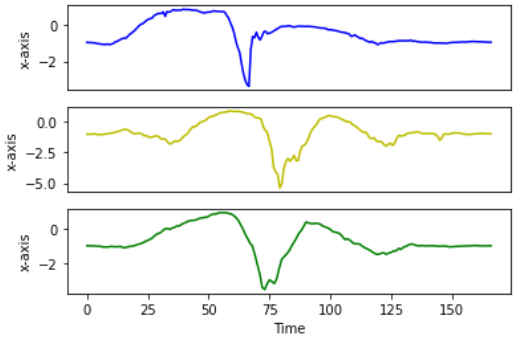
c) Standing



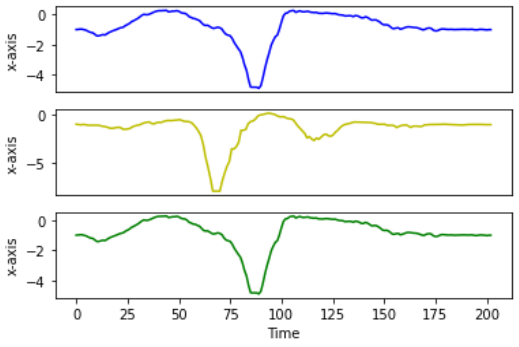
Figures 4 and 5 display 3 samples of raw x-axis signals for three activities across the USC-HAD and UCI-HAD datasets. These figures demonstrate that the data looks similar for these three daily activities - walking, sitting and standing.

Figure 6. Variation in x-axis accelerometer signals for 4 activities and 3 samples each in UTD-MHAD

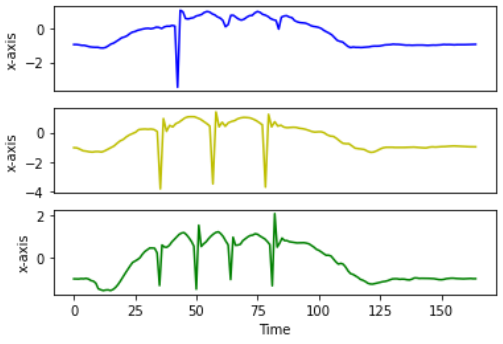
a) Baseball Swing



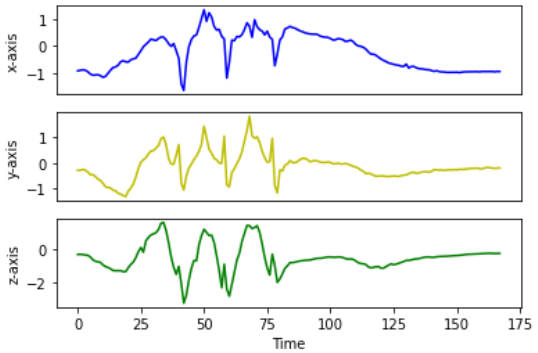
b) Tennis swing



c) Clap

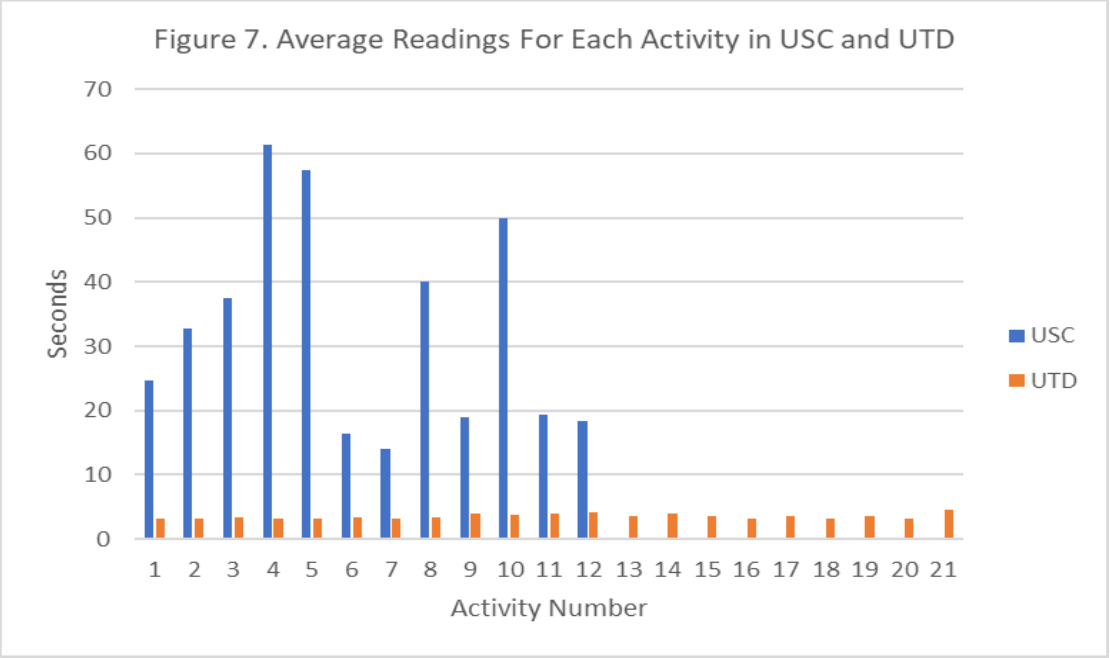


d) knock



On the other hand, it can be observed that the UTD-MHAD activities are distinct across the activities. For instance, between baseball and tennis swing, albeit both characterised as a swing, baseball swing as a sharp pointed drop whereas the drop in the tennis swing has some brief flatness. Between clap and knock, the oscillation in the signals are sharper in clap compared to those seen in knock.

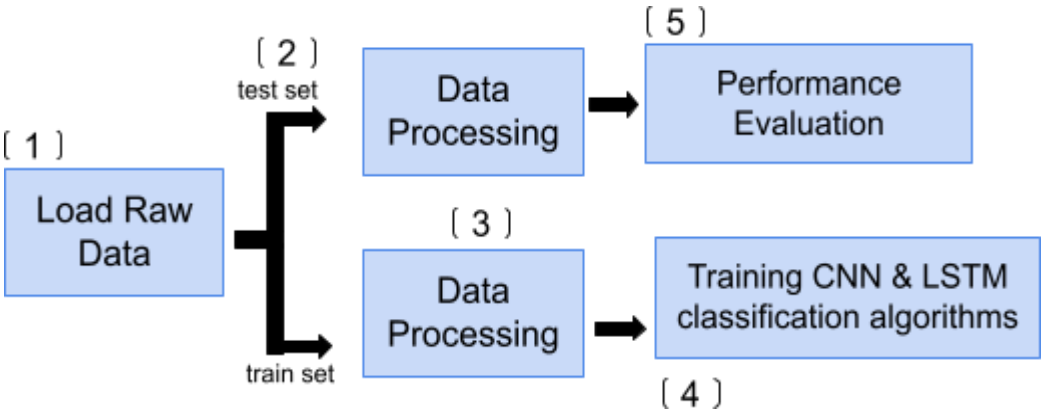
The activities in UTD-MHAD can also be characterised with having instantaneous and short-term movement compared to those activities in USC-HAD and UCI-HAD which have more data in a given time period. This can be further demonstrated in Figures 7 which shows the average readings in seconds for each activity. There is a significantly higher number of readings in each activity in the USC-HAD dataset compared to UTD-MHAD.



4.3 Experimental Design

In order to methodologically answer the research questions above, three separate experiments were conducted. Figure 8 is a summary of the general steps followed.

Figure 8. General Experimental Approach



Load Raw Data

Experiments 1 and 2 : Comparison of Accuracy Scores for Resampling Technique and Window Overlap Rates

The first step was to load the raw data signals for the three datasets. In the first two experiments, we used Leave-One-Subject-Out Cross-Validation to obtain the optimal resampling technique and overlap rates for each dataset.

Experiment 3: Comparison of Different Image Generation Techniques

The DL models need to be able to learn various activity signals from different individuals and predict activities of individuals not seen before. Therefore to generate the training and test sets, the datasets were split by subject. For the UTD-MHAD dataset, a 50-50 training/test split was employed, which was chosen by the authors of the dataset (Chen et. al., 2015). The training subjects were individuals 1, 3, 5, and 7, and the test subjects were individuals 2, 4, 6, and 8. For USC, the split was around 80-20 which was based on the best configuration according to an experiment conducted by Sanchez Guinea et. al. (2022) . The training subjects were individuals 1, 2, 4, 5, 6, 7, 10, 11, 12, 13, and 14, and the test subjects were individuals 3, 8, and 9. As mentioned above, the UCI dataset was already split with a 70-30 ratio.

Data Processing

Experiment 1: Comparison of Accuracy Scores for Resampling Technique

In the UTD-MHAD dataset, each sequence for each individual has a different length. The resampling process allows us to achieve equal lengths for each observation. This is not an issue with the USC-HAD and UCI-HAD datasets therefore this experiment was only conducted on the UTD-MHAD dataset. Resampling was achieved through the resample function within the scipy.signal package.

To understand the impact of resampling, two different resampling choices were tested: no resampling and a resampling size of 128. A window size of 128 was chosen based on two factors - previous research and the default window length of the UCI dataset which cannot be altered. Wang et. al. (2018) found that the optimal window length for HAR lies between 2.5-3.5 seconds. The fixed window length of the UCI dataset is 2.56 seconds which falls under this range. Given a window size of 2.56 seconds and a sampling rate of 50Hz in the UTD-MHAD dataset, this produced 128 readings per window. This was also combined with no window overlap rate to hold the resampling variable constant.

Without resampling, the sample size in the training and test sets varied significantly between 794 to 826 and 91 to 123, respectively. A 128 resampling size produced either 566 or 567 samples for the training set and 62 or 63 samples for the test set. *Figures 9 and 10* demonstrate the effect of the resampling technique for one window.

Figure 9. UTD-MHAD Raw Signals Visualisations With No Resampling

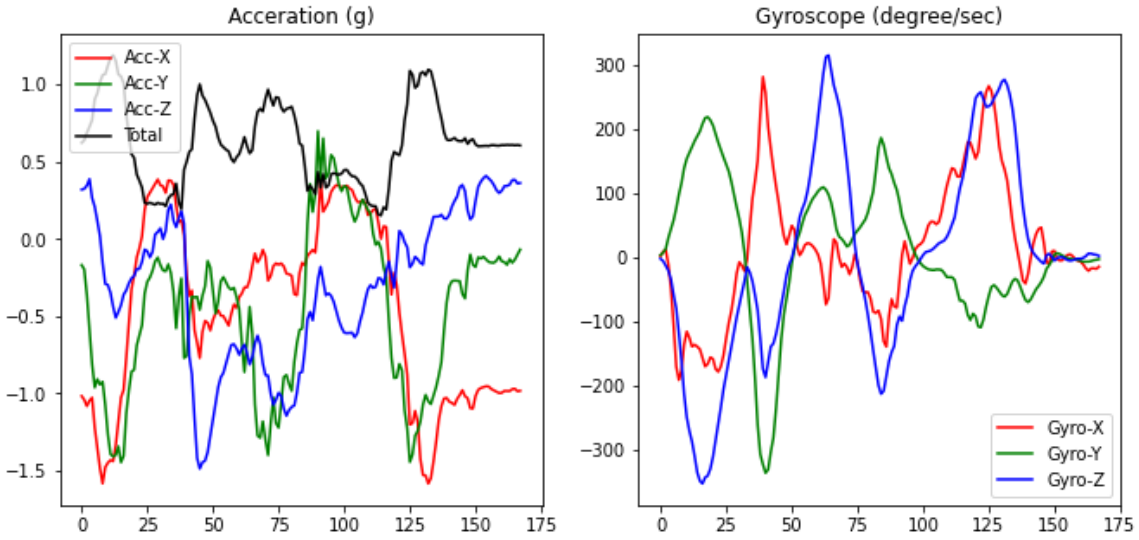
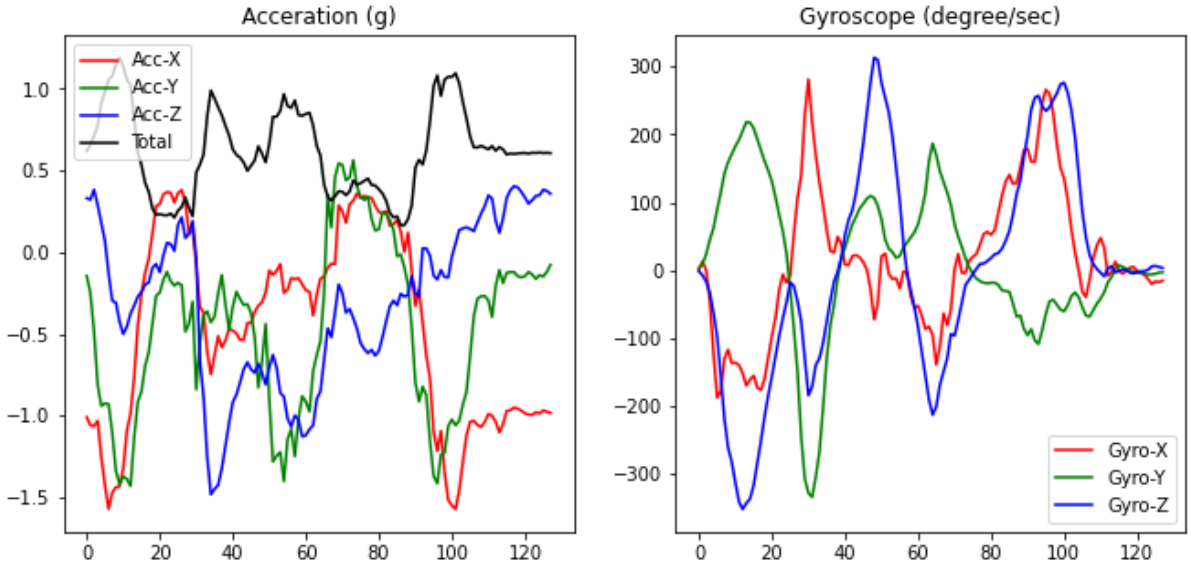


Figure 10. UTD-MHAD Raw Signals Visualisations With Resampling



Experiment 2: Comparison of Accuracy Scores for Window Overlap Rates

To understand the impact of window overlap rates, different overlap rates were tested for the UTD-MHAD and USC-HAD datasets. For UTD-MHAD, the following rates were implemented: no overlap, 50%, and 80%. For USC-HAD, four rates were experimented on: no overlap, 30%, 50% and 70%.. An overlap rate of 50% was chosen given that this is the most widely used rate in the HAR literature. The upper overlap rate was based on the maximum memory capacity of the machine used to train the models. To reiterate, increasing the overlap rate generates more samples for training.

The resampling choice is held constant in UTD-MHAD following the best result in Experiment 1.

In addition, for both the UTD-MHAD and USC-HAD datasets, another sensor reading for the combined acceleration was generated (Noori et.al, 2020). This can be mathematically described as:

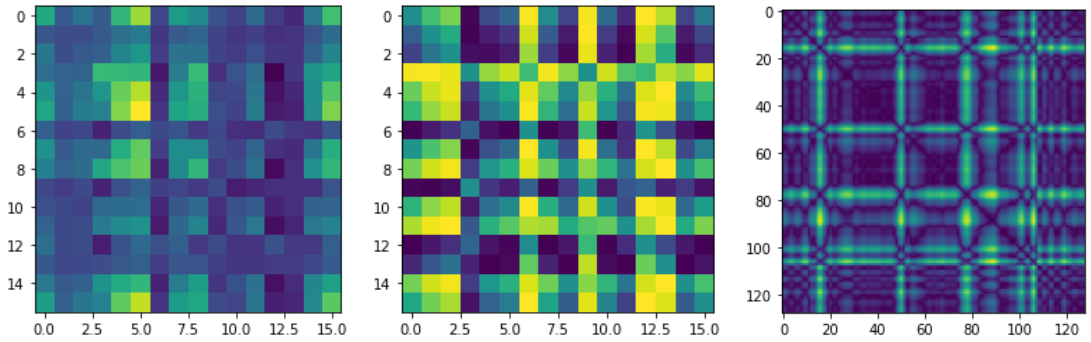
$$Total\ Acceleration(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}$$

where $a_x(t)$, $a_y(t)$, and $a_z(t)$ are each axis in the accelerometer. This is visualised in the solid black line on the left graphs of *Figures 8 and 9*. Total acceleration captures the total movement of the activity with direction having no effect. It is possible that this additional axis can capture more features and thus improve the classifiers. Furthermore, this is equivalent to the total acceleration feature present in the UCI-HAD dataset which can aid comparison between the three datasets. To further improve comparability between datasets, the data was normalised between 0 to 1.

Experiment 3: Comparison of Different Image Generation Techniques

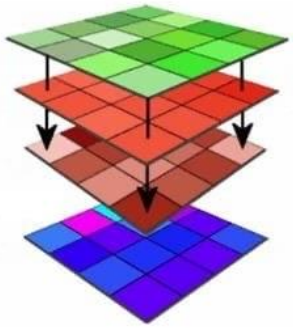
Following the optimal combination from the two preprocessing choices in Experiments 1 and 2, these are applied in Experiment 3. Thereafter, the preprocessed inputs are used to generate images using Gramian Angular Field, Markov Transition Field and Recurrence Plot. These transformations are available from the python package for time series classification called pyts.image. Lastly, the three generated images were stacked vertically together to form one image. Figure 11 displays examples of the type of images produced. Figure 12 visualises how the three different images were stacked together which forms a cuboid that was passed to the CNN model.

Figure 11. Image Generation Samples from UCI-HAD Dataset



a) Markov Transition Field b) Gramian Angular Field c) Recurrence Plot

Figure 12. Stack of Image Generation



Training Phase

For Experiments 1 and 2, the preprocessed inputs will be passed to three baseline models (CNN, LSTM, and Bi-LSTM). During this stage, the models will learn the optimal weights. Sections 4.5 to 4.7 provide further details on the parameters of the baseline models. For Experiment 3, the four generated images will be trained using the baseline CNN model. The best performing CNN model will

be noted and used to train a hybrid CNN+LSTM model. The LSTM variant chosen is based on the best performing model with non-image preprocessed signals as inputs. Therefore the hybrid model consists of the best performing CNN and LSTM models which may not necessarily include image generation techniques. Results were run using a core i7 CPU, Windows 10 OS, GPU RTX 2060 , and a 16 GB RAM.

Performance Evaluation

For Experiments 1 and 2, the average accuracy score across the test subjects was calculated through cross-validation. For Experiment 3, the performance of the models were evaluated on the test set.

4.4 Performance Evaluation

This project will employ accuracy to measure the performance of the different models. It is the most standard metric for classification problems. It calculates the proportion of correct predictions by the total number of samples. The formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where True Positive (TP) - measures the proportion of correctly classified positive class
True Negative (TN) - measures the proportion of correctly classified negative class
False Positive (FP) - measures the proportion of incorrectly classified positive class
False Negative (FN) - measures the proportion of incorrectly classified negative class

4.5 Convolutional Neural Network (CNN) Model

CNNs are mainly used to identify two-dimensional figures that are invariant to displacement, scaling, and other forms of distortion. The network avoids explicit feature extraction and instead implicitly learns from the data. The fact that the image of the input vector can be directly fed into the network avoids the complexity of data reconstruction in the process of feature extraction and classification. Another advantage of CNN, over networks with interconnected neurons, is its local weight sharing characteristic. Through this characteristic, the network is able to learn in parallel and reduce the complexity of the network, which is especially beneficial in multi-dimensional networks.

Table 1. CNN Baseline Model Parameters

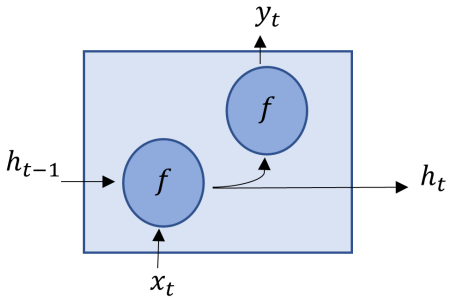
Parameter	Value
Number of convolutional layers	3
Channel	32 for the first layer, 64 for the second layer and 16 for the third layer
Filter Size	5 x 5 for the first layer 3 x 3 for the second and third layer
Activation Function	SeLU(Scaled Exponential Linear Unit)
Normalisation	BatchNorm2d

Drop out	50%
Loss	CrossEntropy
Optimizer	Adam

4.6 Recurrent Neural Network (RNN) and LSTM

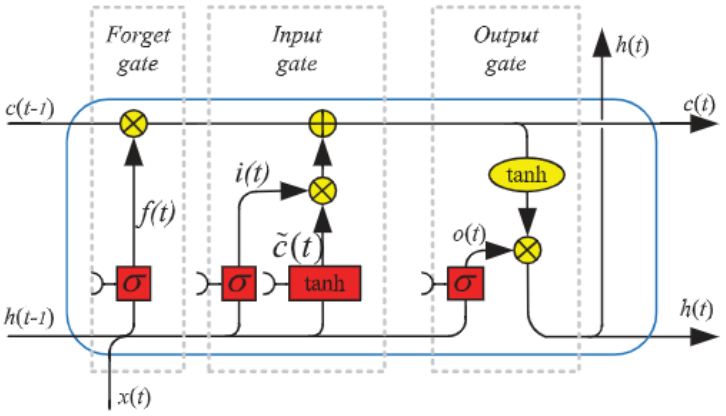
CNN is able to capture the spatial domain of sensor data however, it is unable to uncover the temporal characteristics of the data. With this limitation, RNN has been applied to the HAR problem to provide some significance to the temporal aspects of the data. RNN is a neural network that specialises in processing sequential data. In this network, there is at least one feed-back loop that allows information to persist. Persistence is made possible because the model weights are shared in the network throughout different time points. Weights are shared such that each output from the network is a function of the previous output. The output is then sent back into the network.

Figure 13. Internal Structure of a RNN Hidden Layer



As a result of this process, the gradients decrease exponentially by the time the network arrives at the initial time step. This issue is compounded further after several iterations. Shrinking gradients lead to small weights and biases which is not conducive for the network to learn the optimal weights and biases. This problem is referred to as the vanishing gradient problem. To resolve this problem, an RNN variant has been developed called LSTM. (Zhao et. al., 2018)

Figure 14. LSTM Architecture



Source: Yu et. al. (2019)

LSTM offers a different RNN architecture to avoid the vanishing gradient problem. Figure 14 is the internal structure of the LSTM hidden layer which has additional components such as a forget gate, f_t, an input gate, i_t, an output gate, o_t, and an internal cell state, when compared to the internal structure of the hidden layer in a simple RNN as seen in Figure 13. Through these additional structures, gradients are able to flow through time which overcomes the vanishing gradients problem.

Table 2. LSTM Baseline Model Parameters

Parameter	Value
Number of Hidden Layers	1
Number of Nodes in Each Layer	100
Dropout	50%
Learning Rate	0.0003
Optimizer	Adam

4.7 Bidirectional LSTM

This architecture allows RNNs to be trained in both time directions simultaneously through having both forward and backward layers. The baseline LSTM predicts the current state through the previous information. Bidirectional LSTM improves the baseline LSTM model in its ability to capture previous and subsequent information. (Zhao et. al., 2018)

Table 3. Bidirectional LSTM Baseline Model Parameters

Parameter	Value
Number of Hidden Layers	1
Number of Nodes in Each Layer	100
Dropout	50%
Learning Rate	0.0003
Optimizer	Adam

5. Results

5.1 Comparison of Accuracy Scores for Resampling Technique

For all the models, resampling improved the accuracy score. Table 4 shows the improvements in the CNN, LSTM and Bi-LSTM model are 41%, 4% and 20%, respectively. These findings are consistent with our hypothesis.

Table 4. Average Test Accuracy for UTD-MHAD Dataset

	No resampling, No overlap	128 Resampling, No overlap
CNN Baseline	30.13%	**71.39%**
LSTM Baseline	18.93%	**22.54%**
Bi-LSTM Baseline	26.46%	**46.72%**

5.2 Comparison of Accuracy Scores for Window Overlap Rates

Following the best resampling choice for the UTD-MHAD dataset in Experiment 1, Table 5 reports the accuracy scores on the test set for 3 overlap rates with 128 resampling.

Table 5. Average Test Accuracy for UTD-MHAD Dataset with 128 Resampling

	No overlap	50% Overlap	80% overlap
CNN Baseline	**71.39%**	70.10%	50.14%
LSTM Baseline	18.87%	36.89%	**57.22%**
Bi-LSTM Baseline	46.72%	50.59%	**64.45%**

Given a 128 resampling size and 2.56 seconds sampling time, the CNN models achieved higher accuracy scores when there is no window overlap. In contrast, for the two LSTM models, a larger window overlap rate of 80% led to higher accuracy scores.

Table 6. Average Test Accuracy for USC-HAD Dataset

	0% Overlap	30% Overlap	50% Overlap	70% Overlap
CNN Baseline	**71.05%**	69.83%	68.95%	68.06%
LSTM Baseline	53.00%	54.64%	62.38%	**67.50%**
Bi-LSTM Baseline	64.22%	67.67%	71.15%	**72.27%**

The observations in the UTD-MHAD dataset are similar with the observations in the USC dataset such that the CNN model performs better without window overlap whilst the LSTM models achieve better accuracy scores with a larger overlap rate.

The results from this experiment are not in line with those found in Yang et. al. (2022) such that the optimal overlap rate for the UTD dataset is 65% and 50% for USC HAD. Although our results are not directly comparable as the sampling times used in Yang et. al. (2022) are 1 second and 2 seconds for UTD and USC, respectively.

5.3 Comparison of Different Image Generation Techniques

Following the results from Experiments 1 and 2, these preprocessing choices were chosen and tested on a number of models for image generation techniques. The results from the baseline models are also presented to understand if image generation can improve the CNN baseline model.

Table 7. Test Set Accuracy Scores for UTD-MHAD, USC-HAD, and UCI-HAD Datasets

Model	UTD-MHAD	USC-HAD	UCI-HAD
CNN Baseline	62.86%	83.58%	87.14%
MTF - CNN	64.13%	73.03%	77.71%
GAF - CNN	76.19%	75.46%	77.06%
RP - CNN	46.35%	68.25%	81.74%
MTF+GAF+RP - CNN	75.56%	78.66%	86.63%
LSTM Baseline	38.73%	82.00%	82.46%
Bi-Directional LSTM Baseline	**77.14%**	**84.62%**	91.42%
Hybrid: CNN + LSTM	66.98%	74.52%	**93.69%**

Table 7 presents the performance of 8 models across the three datasets. It is evident that image generation techniques only improved the performance of the baseline CNN model for the UTD-MHAD dataset. The best performing image generation technique was GAF, which achieved an accuracy score of 76.2%. It improved the CNN baseline model by 13.3%. This was not in line with our hypothesis such that the combined image generation technique will yield the best results. However, this result was in line with the findings of Ahmad and Khan (2021) on the same dataset such that GAF outperforms the other two image generation techniques.

On the other hand, for the USC-HAD and UCI-HAD datasets, image generation techniques did not lead to better results compared with the CNN baseline model. For USC-HAD and UCI-HAD, the combined image generation technique outperformed the other image generation techniques. However, these models underperformed the CNN baseline model by 4.9% and 0.5%, respectively.

In terms of the best accuracy for the USC-HAD dataset, the Bi-Directional LSTM model achieved the highest accuracy score at 84.62%. For the UCI-HAD dataset, the hybrid CNN model, which was composed of the CNN baseline model and the Bi-Directional LSTM model, attained the best accuracy score of 93.69%.

6. Discussion

Experiment 1: Comparison of Accuracy Scores for Resampling Technique

Given that resampling techniques improved the performance of all three baseline models in the UTD-MHAD dataset, this preprocessing technique should be included for any signal data with unequal lengths.

Experiment 2: Comparison of Accuracy Scores for Window Overlap Rates

In Tables 5 and Table 6, the CNN models achieved their highest accuracy scores with no window overlap compared with other rates of overlap whilst the LSTM models performed better with a larger rate of overlap. This can be explained by the structure of each network. In CNN, not all the neurons in the upper and lower layers can be directly connected. Instead, they are connected through the convolution kernel which functions as an intermediary. The same convolution kernel is shared in multiple images, and the image can retain the original positional relationship through the convolution operation. The signal of each layer of neurons can only be propagated to the upper layer which means at each moment, the processing of samples is independent.

In view of the limitation that CNN cannot model changes in time series, RNN can complete the processing of time series data. In RNN, the output of neurons can directly affect itself at the next timestamp, and the final result of the network at $(t + 1)$ time $O(t + 1)$ is the result of the interaction between the input and all histories at this moment. Since RNN has the problem of gradient disappearance, the emergence of LSTM solves the problem of long-term dependence well.

Within CNN, the network only pays attention to the image in the current window. In other words, without any window overlap, the network processes the image at the current time point. Once the overlap is applied, it is equivalent to adding a time series to the image. Repeated image features corresponding to different labels will cause ambiguity in model classification and consequently lead to poor results.

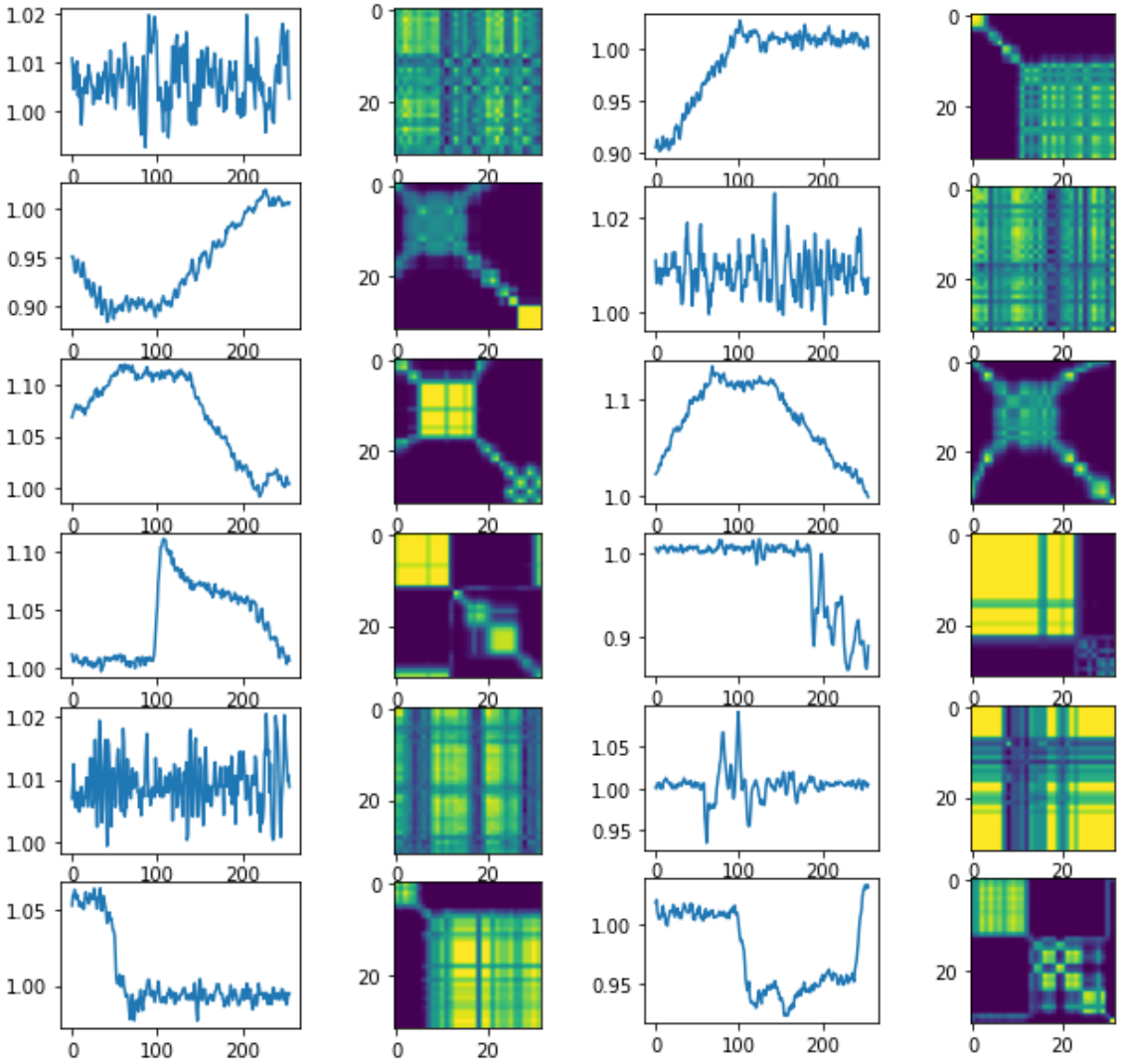
Experiment 3: Comparison of Different Image Generation Techniques

Image generation techniques improved the recognition performance of the CNN baseline model for the UTD-MHAD dataset but not for the USC-HAD and UCI-HAD datasets. The exploratory data analysis conducted in Section 4.2 shows that the UTD-MHAD activities are more specialised activities compared to those contained in the other two datasets which largely covers basic daily activities. The raw signal data plots in *Figure 6* also highlighted distinct features between the UTD-MHAD activities. One potential reason for the success of image generation techniques in UTD-MHAD is that image generation increases the visual differences between the activities such that the distinct features in each activity is further emphasised.

The image generation techniques did not perform as expected in the USC-HAD and UCI-HAD dataset because the raw signal plots between the activities are similar to each other as shown in *Figures 4 and 5*. The USC-HAD dataset consists of relatively basic activities where all axis data for the different activities are very similar. *Figure 15* shows the total acceleration signals from two of the most misclassified activities in CNN-MTF model. The first three images in each activity represent one window from the test set and the next three images come from the training set. It can be seen that there are similar image representations between these two activities especially between the second image in Elevator Up and the third image in Elevator Down. These two activities are closely related to each other which would explain their misclassification.

Figure 15. MTF Images for USC-HAD for the Top Misclassified Activities

Activity 10: Elevator Up Activity 11: Elevator Down

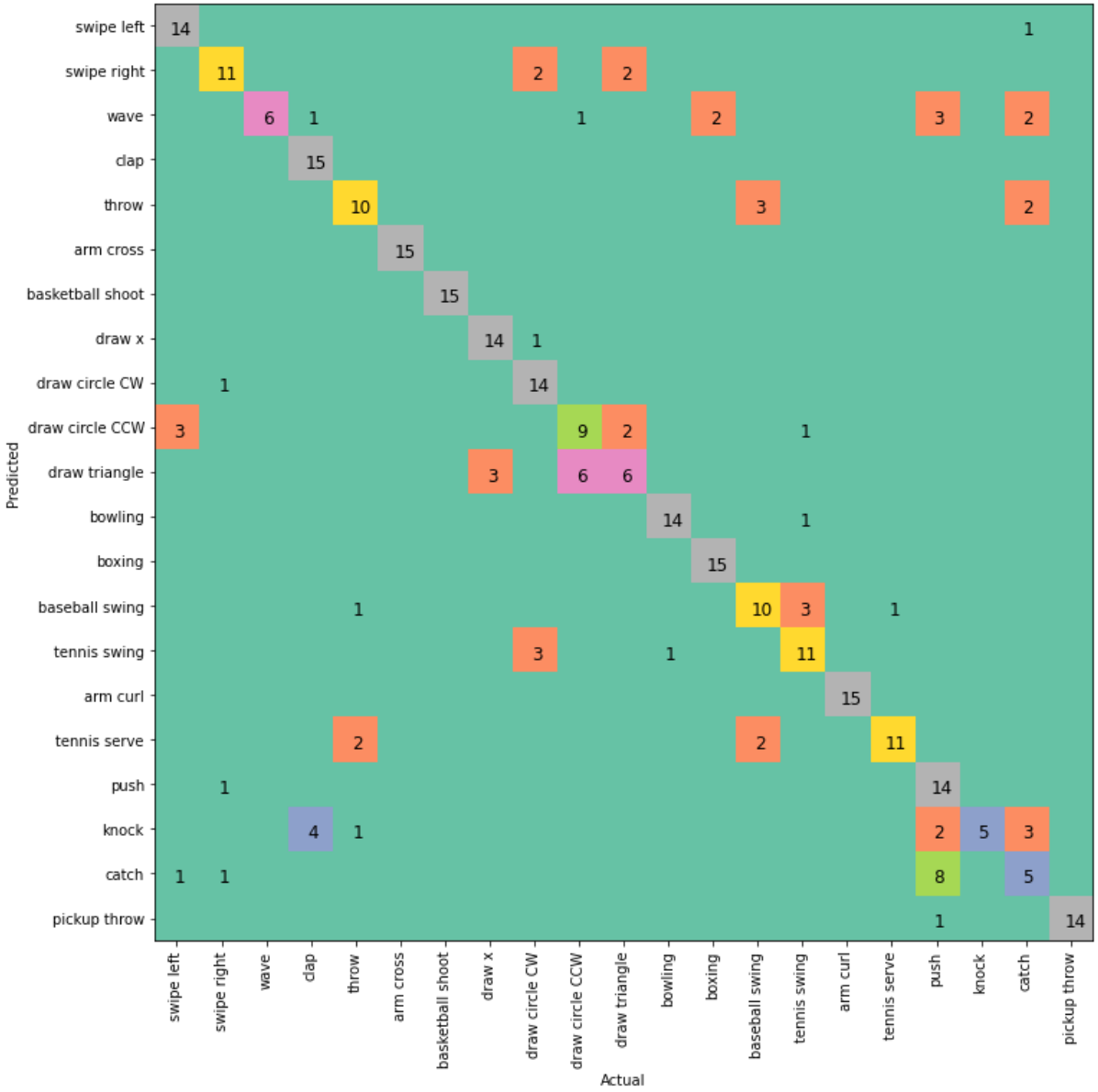


It is possible that image generation is not required to classify basic daily activities given that there are already many features in the signal data which allows a simple neural network to successfully classify the activities. Therefore the addition of image generation techniques simply adds noise to the network.

Although not directly relating to the performance of image generation techniques, it is also significant to note that the Bi-directional LSTM models for UTD-MHAD and USC-HAD and a CNN-LSTM hybrid model for the UCI-HAD dataset outperformed the CNN models. This indicates that the temporal variations in the HAR problem is an important consideration. When comparing the performances across the three datasets, UCI-HAD and USC-HAD outperformed UTD-MHAD because as seen in Figures 4 to 6, there are more temporal differences in the former two datasets than the latter. The activities considered in the UTD-MHAD dataset have little variation in the first and last quartile of the time period compared to the other two datasets.

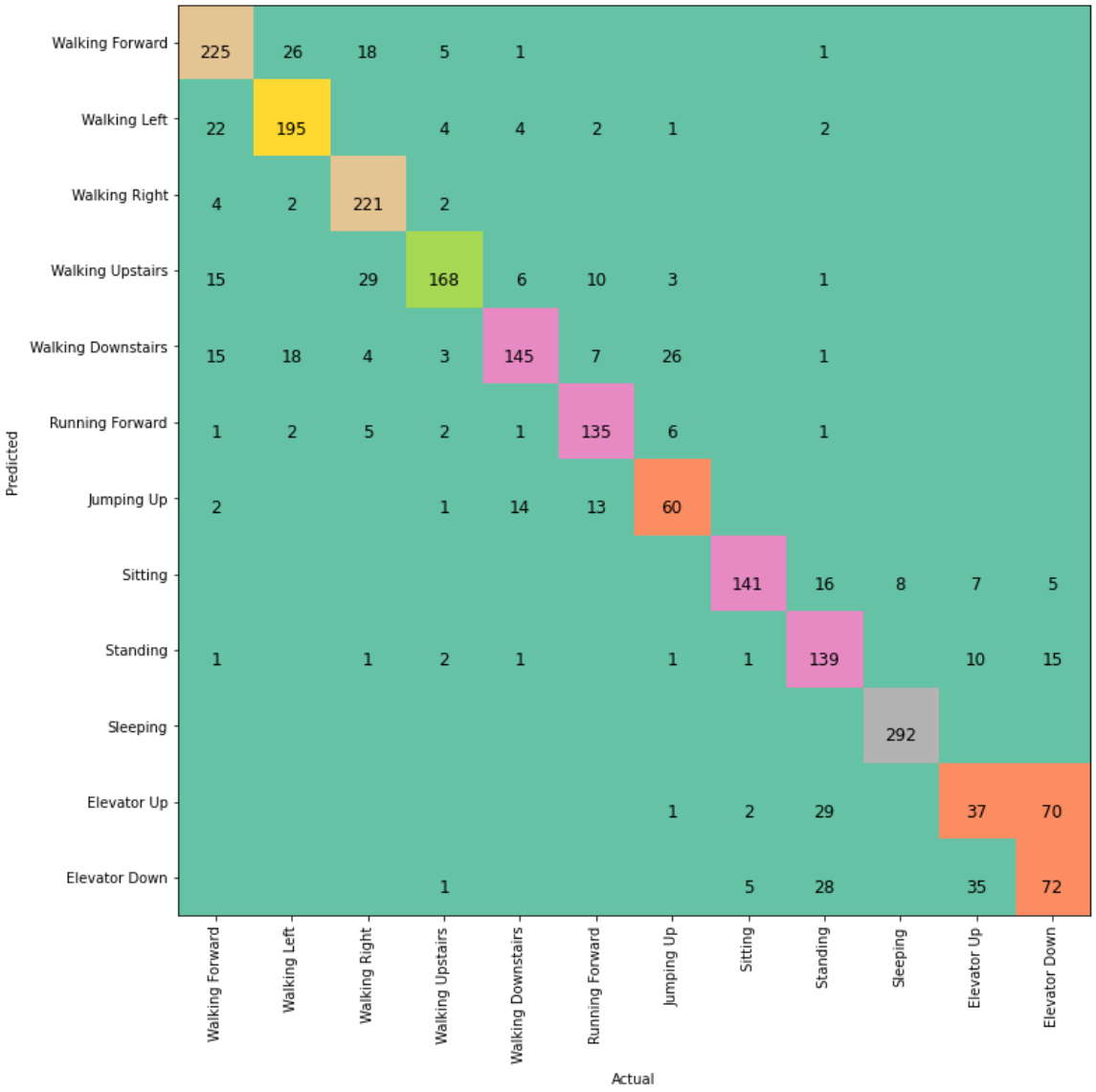
Lastly, to visualise the performance of the best models for each dataset, the confusion matrices are presented below.

Figure 16. Bi-Directional LSTM Model Confusion Matrix for UTD-MHAD



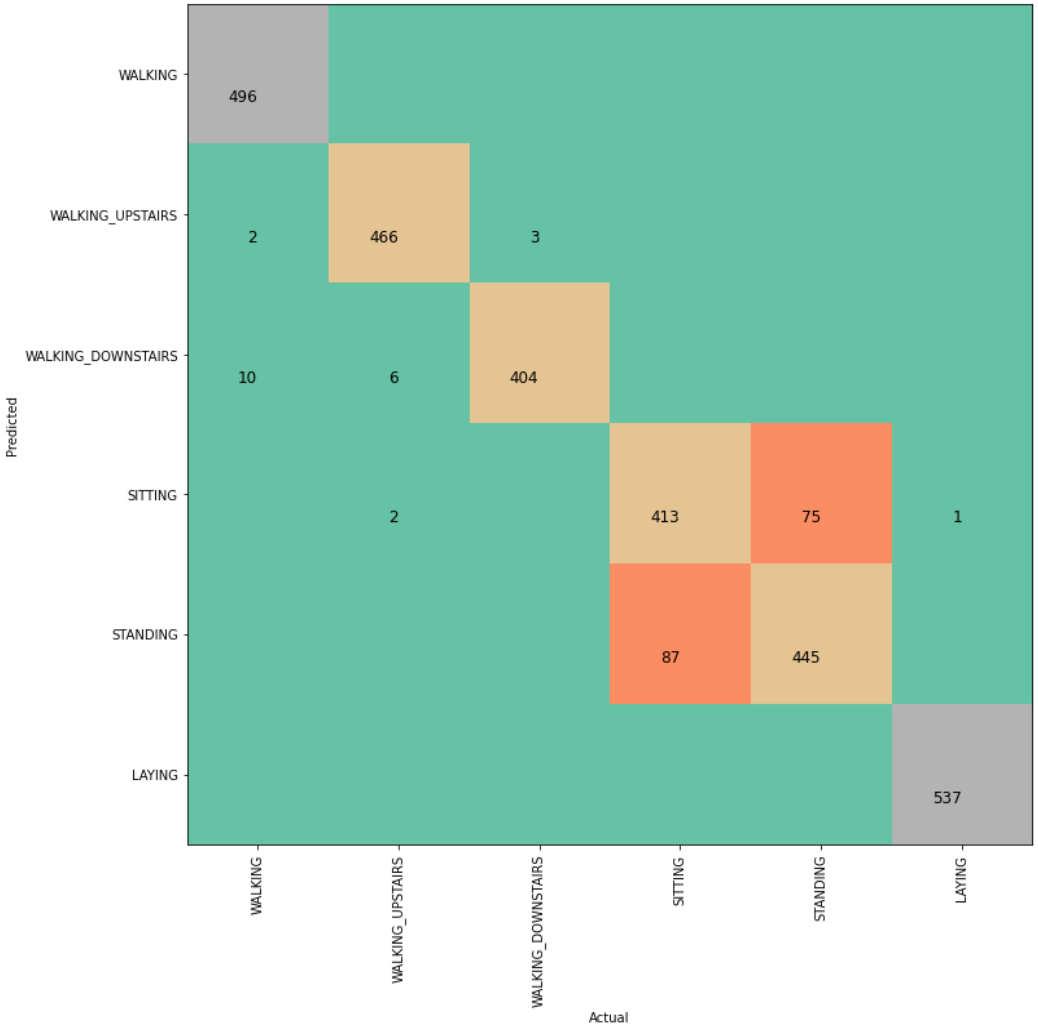
The Bi-Directional LSTM model achieved the highest accuracy score of 77.14% for the UTD-MHAD dataset. The most misclassified activities are push and catch with 32% of push labels were incorrectly classified as catch.

Figure 17. *Bi-Directional LSTM Model Confusion Matrix for USC-HAD*



For the best model in USC-HAD, the most misclassified activities are Elevator Up and Elevator Down. 39% of Elevator Up labels were predicted as Elevator Down whilst 43% of Elevator Down labels were predicted as Elevator Down. These two activities are essentially the same except for the temporal information which is understandable why they are the most misclassified. An important detail which can also contribute to their misclassification is the unbalanced dataset where these two activities have been underrepresented in the sample with, which only form 5.8% and 5.9% compared to more represented activities that form around 9% - 13%.

Figure 18. *Hybrid CNN-LSTM Model Confusion Matrix for UCI-HAD*



With regards to the best model for the UCI-HAD dataset, Figure 18 shows that the most misclassified activities are Standing and Sitting. Specifically, 14% of the Standing labels were misclassified as Sitting and 17% of the Sitting labels were incorrectly predicted as Standing. This misclassification is largely due to the similarities in the data across some axes and only the orientation can differentiate these actions.

7. Conclusion and Future Improvements

This project explored the effects of different preprocessing techniques such as resampling, window overlap rates, and image generation techniques on three deep learning models: CNN, LSTM, and Bi-directional LSTM and three publicly available HAR datasets: UTD-MHAD, USC- HAD, and UCI-HAD. In the first experiment, it is evident that resampling has a positive effect on all the deep learning models given a 128 resampling size. A limitation of this experiment is that only one resampling size of 128 was tested. Future works could look at experimenting on other resampling sizes such as 64 and 256. In the second experiment, given a 128 resampling size and 2.56 seconds sampling time, the CNN model achieves higher accuracy scores when there is no window overlap. In contrast, for the two LSTM models, a larger window overlap rate of 80% for the UTD-MHAD dataset and 70% for the USC-HAD dataset produced higher accuracy scores. Due to time and computational resource constraints, this project experimented with only 3 overlap rates on the UTD-MHAD dataset and 4 overlap rates on the USC-HAD dataset. For future experiments, it would be beneficial to test more overlap rates to fully identify the optimal overlap rate for each dataset.

For the image generation experiment, the results were varied such that image generation techniques improved the performance of the CNN baseline model for the UTD-MHAD dataset but this was not the case for the USC-HAD and UCI-HAD datasets. Given that the activities in the USC-HAD and UCI-HAD datasets cover basic activities where variations between activities can only be meaningful over longer time windows, larger window sizes need to be considered for image generation techniques to work in these two datasets. In this experiment, the time window was fixed at 2.56 seconds which is not sufficient. Future experiments should test time windows of around at least 10 seconds. Another possible improvement, in particular for the UTD-MHAD dataset, is to train the static and dynamic activities separately which can be followed by feature level fusion. This approach can be beneficial for the UTD-MHAD dataset as it contains a large number of activities.

As a general improvement to all the experiments, the imbalanced activity distribution for USC-HAD and UCI-HAD datasets, as shown in Figures 2 and 3, needs to be addressed. We also did not perform hyper-parameter tuning for many parameters such as the learning rate, drop-out rate and number of convolution layers. These additional steps may improve the accuracy of the models.

Bibliography

Anguita, D., Ghio, A., Oneto, L., Parra Perez, X. and Reyes Ortiz, J.L., (2013). A public domain dataset for human activity recognition using smartphones. In Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning (pp. 437-442). [Online] Available at: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

Ahmad, Z. and Khan, N. (2021). Inertial Sensor Data to Image Encoding for Human Action Recognition. IEEE Sensors Journal, 21(9), pp.10978–10988.

Chernbumroong, S., Cang, S., Atkins, A. and Yu, H. (2013). Elderly activities recognition and classification for applications in assisted living. Expert Systems with Applications, 40(5), pp.1662–1674.

Chen, C., Jafari, R. and Kehtarnavaz, N. (2015). UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/7350781 [Accessed 26 Jun. 2020].

Chen, Y. and Xue, Y. (2015). A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. 2015 IEEE International Conference on Systems, Man, and Cybernetics. [online] Available at: https://ieeexplore.ieee.org/document/7379395/ [Accessed 16 March 2022].

Debayle, J., Hatami, N. and Gavet, Y. (2018). Classification of time-series images using deep convolutional neural networks. Tenth International Conference on Machine Vision (ICMV 2017).

Foerster, F., Smeja, M. and Fahrenberg, J. (1999). Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. Computers in Human Behavior, 15(5), pp.571–583.

Gu, F., Chung, M.-H., Chignell, M., Valaee, S., Zhou, B. and Liu, X. (2021). A Survey on Deep Learning for Human Activity Recognition. ACM Computing Surveys, 54(8), pp.1–34.

Han, B., Zhang, H., Sun, M. and Wu, F. (2021). A New Bearing Fault Diagnosis Method Based on Capsule Network and Markov Transition Field/Gramian Angular Field. Sensors, 21(22), p.7762.

Jiang, W. and Yin, Z., (2015). Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks. In: Proceedings of the 23rd ACM International Conference on Multimedia. pp. 1307-1310.

Jordao, A., Nazare, A., Sena, J. and Robson Schwartz, W. (2019). Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-Art. [online] Available at: https://arxiv.org/abs/1806.05226 [Accessed 16 March 2022].

Lara, O.D. and Labrador, M.A. (2013). A Survey on Human Activity Recognition using Wearable Sensors. IEEE Communications Surveys & Tutorials, [online] 15(3), pp.1192–1209. Available at: https://ieeexplore.ieee.org/document/6365160 [Accessed 23 Mar. 2022].

Motion Workshop Company (2021). MotionNode Miniature Inertial Measurement Unit. [online] www.motionnode.com. Available at: https://www.motionnode.com/ [Accessed 20 Apr. 2022].

Noori, F.M., Riegler, M., Uddin, M.Z. and Torresen, J. (2020). Human Activity Recognition from Multiple Sensors Data Using Multi-fusion Representations and CNNs. ACM Transactions on Multimedia Computing, Communications, and Applications, 16(2), pp.1–19.

Panwar, M., Biswas, D., Bajaj, H., Jobges, M., Turk, R., Maharatna, K. and Acharyya, A. (2019). Rehab-Net: Deep Learning Framework for Arm Movement Classification Using Wearable Sensors for Stroke Rehabilitation. IEEE Transactions on Biomedical Engineering, 66(11), pp.3026–3037.

Sanchez Guinea, A., Sarabchian, M. and Mühlhäuser, M. (2022). Improving Wearable-Based Activity Recognition Using Image Representations. Sensors, 22(5), p.1840.

Tufek, N., Yalcin, M., Altintas, M., Kalaoglu, F., Li, Y. and Bahadir, S.K. (2020). Human Action Recognition Using Deep Learning Methods on Limited Sensory Data. IEEE Sensors Journal, [online] 20(6), pp.3101–3112. Available at: https://www.research.manchester.ac.uk/portal/files/149703187/activity_journal_4_1_.pdf [Accessed 17 March 2022].

Wang, Z. and Oates, T. (2015). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In Proc. Workshops 29th AAAI Conf. Artif. Intell., 2015, pp. 1–8.

Wang, G., Li, Q., Wang, L., Wang, W., Wu, M. and Liu, T. (2018). Impact of Sliding Window Length in Indoor Human Motion Modes and Pose Pattern Recognition Based on Smartphone Sensors. Sensors, 18(6), pp.1965–1985. doi:10.3390/s18061965.

Wang, J., Chen, Y., Hao, S., Peng, X. and Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters, [online] 119, pp.3–11. Available at: https://www.sciencedirect.com/science/article/abs/pii/S016786551830045X [Accessed 16 March 2022].

Xu, H., Li, J., Yuan, H., Liu, Q., Fan, S., Li, T. and Sun, X. (2020). Human Activity Recognition Based on Gramian Angular Field and Deep Convolutional Neural Network. IEEE Access, 8, pp.199393–199405.

Yang, P., Yang, C., Lanfranchi, V. and Ciravegna, F. (2022). Activity Graph based Convolutional Neural Network for Physical Activity Recognition using Acceleration and Gyroscope Data. IEEE Transactions on Industrial Informatics, [online] pp.1–1. Available at: https://ieeexplore.ieee.org/abstract/document/9681285 [Accessed 10 March 2022].

Yu, Y., Si, X., Hu, C. and Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 31(7), pp.1235–1270.

Zhang, M. and Sawchuk, A.A. (2012). USC-HAD. Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12, [online] pp.1036–1043. Available at: https://dl.acm.org/doi/abs/10.1145/2370216.2370438 [Accessed 12 Apr. 2022].

Zhang, S., Li, Y., Zhang, S., Shahabi, F., Xia, S., Deng, Y. and Alshurafa, N. (2022). Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. Sensors, 22(4), p.1476.

Zhang, G., Si, Y., Wang, D., Yang, W. and Sun, Y. (2019). Automated Detection of Myocardial Infarction Using a Gramian Angular Field and Principal Component Analysis Network. IEEE Access, 7, pp.171570–171583.

Zhao, Y., Yang, R., Chevalier, G., Xu, X. and Zhang, Z. (2018). Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors. Mathematical Problems in Engineering, 2018, pp.1–13.