Human Activity Recognition using Accelerometer Data

Dashen Govender [218031968]

June 2021

Abstract

In this investigation, we will research and gather information on Human Activity Recognition (HAR) to highlight key areas of the topic, the scope of HAR, the impact of HAR in different works and our work, and the purpose of our investigation. We will classify human activities produced by a mobile accelerometer and derive precise conclusions based on our experiment with smartphones. Different activities performed using a smartphone, and the results they give are target explanations in this research. The problem of our research lies in the ability of smart models to classify this data. We will be using Convolutional Neural Networks and Recurrent Neural Networks to see how capable these perform with accelerometer data. To solve our problem, we will have to construct a confusion matrix out of our models and determine graph results, then relate these results with other artificial intelligence classification works, which may include digital image processing or natural language processing in the use of machine learning. It is of great importance that the reader becomes well informed in this area of expertise. The results achieved here have shown us how specific activities affect the performance of our models. Our findings have determined the more suitable model to be used and the effect of an activities shape, direction and distance on both models.

1 Introduction

Human Activity Recognition (HAR) is a challenging time series classification task that predicts a person's movements based on sensor data, deep domain expertise, and methods from signal processing. Signal processing correctly engineers a feature from raw data to fit a machine learning model. Devices such as smartphones and smartwatches are ubiquitous and context-aware. Devices like these provide a gateway for human-centric computing and allow widespread monitoring of people's physical activities and daily routines. [1]

Sensor-based recognition applications include context awareness, human-computer interactions, smart homes, healthcare, and security control. These application areas can be tackled using accelerometer or gyroscope data from smart devices. In particular, an accelerometer captures the linear acceleration, while the gyroscope measures the body's rotational motion. The challenge of sensor-based classification is to ensure the welfare of users who have devices that track their movement. [1]

Significant advancements in the technology and Artificial Intelligence sector have been made with the use of HAR. By using a more dynamic approach to reading data, technologists have opened a window of opportunities. We can use these opportunities to interpret a humans behaviour based on sensor and video observation data and knowledge of the context of the environment. A HAR system, in general, reflects a cognitive process for learning and understanding based on human actions. [2]

Our investigation aims to discover the challenge of classifying activities and how different motions effect classification, as well as the effectiveness of the subjects metrics on the HAR system. We will have a look at different variations of classification. Our focus will be on outdoor activities that include moving downstairs, jogging, sitting, standing, moving upstairs, and walking. Our HAR research task is crucial as it may provide information about accelerometer data that can be used to improve smart systems or verify the limitations of smart systems. Limitations may include the effectiveness of smart devices with accelerometers' in different environments, such as the wild, a home, a mall etc. [1]

Our paper will discuss related works, methods and techniques used to classify the features used, the experimental results obtained from testing a dataset containing accelerometer data, and any future studies that may involve our work. In particular, our research seeks to establish whether accelerometer data provides good CNN or RNN (with LSTM) readings for classifying activities. We will contrast both these classification techniques in our experiment. Our work is different to other related works as we focus on two different (ML) Artificial Neural Network techniques that have not been used together or compared in the field of ANN in previous studies to deal with accelerometer data. Our investigation is necessary as it will help determine the downfalls of gravitational effects on a smart device. It will also show us how effective our smart models are at predicting different activities and their limitations. [3], [4]

2 Literature Review and Related Works

In work [5], tri-axial accelerometers were used to detect everyday activities. The accelerometers were placed on the chest, wrist, lower back, hip, thigh and foot. 8 healthy males participated in the study; their ages were from 24 to 33. Activities included walking, running on a motorized treadmill, sitting, lying, standing and walking down

and up some stairs. A total of 370,000 samples were used, with 50,000 samples for each activity. Acceleration data was sampled at 50 Hz using an accelerometer with a range of ±6 g. The machine learning models that were applied to this dataset were Decision Trees (DT, J48), Naïve Bayes (NB), Neural Network (NN) and Support Vector Machine (SVM).

In this study, when reviewing different placements of the accelerometer on the body, SVM provided better results than DT. NB and NN provided better results than DT. SVM had the highest average accuracy of 96.67%. DT had an average accuracy of 94.18%, NB an average accuracy of 94.77% and NN an average accuracy of 95.74%. In this study, we see that the hip location has the highest accuracy reading of 97.75% when using NN. Since SVM's can be implemented in real-time and work well with noisy datasets, it was used for subsequent testing. The hip using SVM achieved the highest accuracy of 97.81%, whereas the foot and wrist achieved the lowest accuracy of 95.63% and 95.88%, respectively. Using the left hip, Lying, Running, Sitting and Standing were classified with 100% accuracy using NN. Throughout all accelerometer locations, stairs up and stairs down seem to be classified with the least accuracy. In this investigation, sensors were also combined to discover if using multiple sensors made a difference. It was discovered that there was no significant difference between using multiple sensors. There was an average of 97.66% of correctly classified instances when 3 sensors were used, the highest. There was an average of 96.6% when 1 sensor was used, the lowest. The most optimal number of sensors used here was two sensors. Using fewer sensors provided better results.

In work [6], the performance of Smartphone sensor behaviour on human activity recognition is investigated. This work analysis was conducted using 27 681 sensory samples from 10 male subjects. Sensory data included a 3-axis accelerometer, a gyroscope, and a magnetic field sensor. A sampling rate of 20 Hz was used. The daily activities that were performed included descending stairs, ascending stairs, walking, jogging, and jumping. Time, frequency and wavelet domain features were extracted from the data. 5 positions of carrying a smartphone were considered: right upper arm, right hand, right jacket pocket, right trousers pocket and waist. Three classification techniques were used in this study; these were Nearest Neighbour (NN) (K = 3), Random Forests (RF) and Support Vector Machines (SVM).

For the NN approach, K was set to 5. RF is an ensemble method. The ensemble method used here was bagging, which involved bootstrapping the dataset and selecting a majority vote of the outputs. 200 trees were used for ensemble learning. A one-versus-one approach was used for SVM since a one-versus-rest approach will lead to inconsistent results and harm the symmetry of the problem. Three conditions in this paper were investigated:
A personalized model: the training and testing data are all from the same subject (one-to-one model).

Generalized model 1: training data comes from all the subjects, and testing data comes from only 1 subject (all-to-one model).
Generalized model 2: the testing data are from only one subject, and the training data are from the rest (rest-to-one model).

It is important to note that weight, height, gender and physical condition will affect movement. The data was divided into 10 parts for the one-to-one model and split into testing and training sets. For the all-to-one, the subjects' were divided into 10 subsets, and each subset was divided into 10 parts, split into training and testing sets. For the rest-to-one model, the data parsing is the same as all-to-one, except the raining set excludes the test subject's data.

From the results of the one-to-one approach, we see that when the user holds the smartphone in hand, he exhibits complex movements such as anterior-posterior and left-right movements. The movement patterns of the thigh are much simpler and periodic. The thigh mainly moves in anterior-posterior and vertical directions. The sensor placed here gives the best accuracy results for walking, jogging, and jumping. RF's have shown to be most effective at classifying because of their ability to perform feature selection automatically and have fewer parameters to set. The all-to-one and rest-to-one model seems to perform better, especially the all-to-one, because it has more training data, including personalized data. The F-Score for a personalized and general model can reach 95.95% and 96.26%, respectively.

When the gyroscope and accelerometer sensors were combined, they exhibited much better accuracy, and a gyroscope performed better than an accelerometer. The results suggest that the choice of sensors should depend on the specific task; on the contrary, it is better to combine multiple types of sensors. This work had some limitations: The dataset used was impoverished, more subjects could have been used; physical conditions affected the movements, e.g., as you climb stairs, you get progressively slower, the experiment was environment-controlled; more practical applications could have been used; an offline dataset was used, an online system remains to be explored.

In work [7], deep activity recognition (DAR) models were produced using triaxial accelerometers. Accelerometer data in this dataset was recorded as follows: samples are of the form $r_t = r_t^* + w_t$ (t (time)=1,2,...).r_t is the acceleration in the x, y, and z direction. They are floating-point values. They are less than B, where $B > 0$. B is the number of gravitational units. $1g = 9.8m.s^2$. On a flat surface, we have 1g. r_t^* is a vector that contains 3-axial noiseless acceleration readings. w_t is a noise vector of independent, zero-mean Gaussian random variables with variance σ_w^2. These are then put into channel frames s_t^x, s_t^y and s_t^z. Based on these, an occurrence of an activity y_t is obtained.

The proposed solution in this work used 3 datasets:
The WISDM Actitracker dataset: This dataset contains 1,098,213 samples of one tri-

axial accelerometer sampled at a rate of 20 Hz. The data samples belong to 29 users who performed 6 distinct activities. Walking, jogging, sitting, standing, and climbing stairs. Mobile phones were used.
Daphnet freezing of gait dataset: This dataset was used to demonstrate the healthcare applications of DAR models. Patients with Parkinsons disease were used to collect data samples. The accelerometers were placed on the ankle, upper leg, and trunk with a sampling frequency of 64 Hz. The goal is to detect whether a patient is labelled "freezing" or "no freezing". 1,140,835 samples from 10 people were used.
Skoda checkpoint dataset: This dataset has 10 distinctive activities that belong to a car maintenance scenario in typical quality control checkpoints. The sampling rate is 98 Hz. Only one accelerometer node is used as opposed to 20 because of cost and convenience.

Three performance metrics were used for binary classification (Daphnet dataset): sensitivity, specificity, and accuracy. For multiclass classification (WISDM Actitracker and Skoda checkpoint datasets), the average recognition accuracy is found. In the results obtained, the DAR models show significant accuracy improvement over conventional methods. For example, it improves accuracy by 6.53% over MLPs and 3.93% over ensemble learning on the WISDM Actitracker dataset. The results of DAR models in this work can be summarized as follows:

1) Deep versus shallow models: this experimentation shows that using DAR models greatly enhances the recognition accuracy compared with conventional shallow models. Furthermore, DAR models automatically learn meaningful features and excuse the need for hand-engineering features, e.g., statistical features, in state-of-the-art methods. Using more layers with fewer neurons is more convenient for accuracy. E.g., 4 layers with 500 neurons are better than 1 layer with 2000 neurons. Overcomplete representations are also efficient, which means that there are more neurons than input layers.

2) Semi-supervised learning (SSL): Most data accumulated is unlabelled. This occurrence motivates SSL (generative training) to fit better activity classifiers, which is important for weight tuning and optimization. Generative (G) pretraining with discriminative (D) training is shown to have better accuracy than just discriminative training as it better generalizes the solutions, which is suitable for many neural layers and not just one layer. For 5 layers of G and D training, we get 97.85% accuracy; for D training only, we get 96.51%.

3) Spectrogram analysis: Multifrequency, aperiodic, and fluctuating signals are generated from accelerometers, which complicate the activity recognition using time series data. Spectrogram signals perform better than raw acceleration data to capture variations in the input data.

4) Temporal Modelling: A hybrid approach and Hidden Markov model (DL-HMM)

was used. 3 layers of 1000 neurons were used for the model. For the 10 activities on the Skoda dataset (Node ID 16), the recognition accuracy was 89.38%, improving by 3.38% over the HMM method presented by (Zappi et al. 2008). The hybrid DL-HMM achieves near perfect recognition accuracy of 99.13%. The experiments show that a DL-HMM outperforms HMM-based methods for temporal activity recognition.

In work [8], on-body accelerometer wearable sensors are used to classify human activities. This work uses a supervised approach to learning data. Physical activities performed in this study included: static postures, such as standing, sitting, lying or dynamic motions, such as walking, running, stair climbing, and cycling. This paper distinguishes between primitives, namely elementary activities, and composite activities like sitting-standing-walking-standing-sitting. The acceleration data was sampled at 76.25 Hz. This data was acquired from 5 bi-axial accelerometers located at the hip, wrist, arm, ankle, and thigh. Feature vectors were built with 512 samples. 13 subjects were randomly selected for this study.

In this study a distinct classifier was trained for each individual subject. The following single-frame classification algorithms were trained and tested using data frames from the reduced dataset: Probabilistic approach: Naïve Bayesian (NB), Gaussian Mixture Model (GMM), Logistic Classifier, Parzen Classifier; Geometric approach: Support Vector Machine (SVM), Nearest mean (NM), k-NN, ANN (multilayer perceptron); Binary decision: Binary decision tree (C4.5). A cHMM sequential classification algorithm was used. This algorithm was a Q-state cHMM with continuous Gaussian emissions (Q (Number of primitives) = 7). A first phase training and second phase training was run. A Transition Probability Matrix (TPM) was made to yield results. Lying, cycling and sitting were classified correctly with more than 80% accuracy. The rest were below 80%, and it seems that running was classified the worst, with 40% accuracy.

The training set for single-frame classifiers is composed of K frames per class and per subject. K=7 was chosen because of its convenience. The number of Gaussian components of the mixture is M = 1, either in the GMM or the cHMM-based classifiers. In the single frame classifier performance, GMM achieved an accuracy of 92.2%, the lowest, and NM achieved an accuracy of 98.5%, the highest. We will take note of ANN, which achieves an accuracy of 96.1%. Without a rejection of spurious data, a classification accuracy of 73.3% is attained. With the rejection of spurious data, a classification accuracy of 99.1% is attained.

All tested classifier accuracies for classifying the seven primitives were remarkably high in this study. Markov modelling proved effective in this paper. Notice that the cHMM-based sequential classifier performs systematically better than its simple single-frame GMM counterpart. (99.1% vs 92.2%). NM, which achieved the highest, indicates the relevance of exploiting the statistical knowledge about the human motion dynamics that is "trapped" within the Markov chain. The cHMM-based sequential classifier was

split into two phases in this paper. Using both phases gave better results, opposed to using just the first phase. This recipe is useful because it effectively copes with the size limitations of the training set.

3 Methods and Techniques

3.1 Analysing The Data

For our work, we will be using the WISDM_Act_v1.1_raw.txt dataset. [9]–[11] This dataset contains six different activities performed by 36 users. The activities performed are walking, jogging, upstairs, downstairs, sitting and standing. The format of a line of data is as follows: [user],[activity],[timestamp],[x-acceleration],[y-accel],[z-accel]; The data has been recorded using phones with tri-axial accelerometers, including gravitational pull. The acceleration ranges from -20 to 20, where $10 = 1g = 9.81m/s^2$. The sampling rate is 20 Hz (1 sample every 50ms).

We are given 1,098,207 examples of data. We will first need to read the data into Pandas data frame. You will see that the data has some lines that contain two data recordings; therefore, we will need to read the data and eliminate any data that gives us errors when we process it. We created a list called processedList to contain all the samples of data. We will need to create column headers and then finally read the data into Pandas data frame for our next step.

For our next step, we will have to specify the acceleration columns as float values. Then we will create a balanced dataset with activity labels and acceleration data. Since the minimum amount of samples for an activity is 3555, we will take only 3555 samples from each activity to have a balanced dataset. After this step, we create a scaler that has labels for each activity.

We will need to prepare frames of 4 seconds for every acceleration; this includes x, y and z. For this preparation, we will need a frequency of 20 Hertz, a frame size of 80 Hertz, and a hop size of 40 Hertz. We consider the label that occurs the most times out of the selective sample of data for our label. Frames are our feature spaces, and labels are our target variables.

At this stage, we can create a testing and training split. We will use 20% of the data as our test set and 80% of the data as our training test. We will now need to feed this data into a 2-dimensional convolutional neural network. To do this, we need a 3-dimensional input of the data; we will therefore reshape the data to suit this requirement. Our CNN has four layers, 1 input layer and 3 hidden layers. In our first layer, we use 16 nodes and a 2x2 kernel with 'relu' activation. We dropped 10% of neurons in our first layer. In our second layer, we used 32 nodes with a 2x2 filter and dropped out

20% of neurons. In between the Conv2D layers and the dense layer, there is a 'Flatten' layer. Flatten serves as a connection between the convolution and dense layers. We will now use Dense layers. We use Dense layers for outputting layers.

The first dense layer uses 64 nodes with activation 'relu', then 50% of neurons produced from that layer is dropped out. Finally, the last dense layer uses six nodes, and the activation is 'softmax'. Softmax makes the output sum up to 1, so the model can interpret the output as probabilities. The model will then make its prediction based on which option has the highest probability. After this, we run ten epochs on the training dataset. An epoch indicates the number of batches of a training dataset passed to a machine-learning algorithm to obtain a high validation accuracy depending on the number of passes. We then, proceed to draw a graph for the data and plot a confusion matrix.

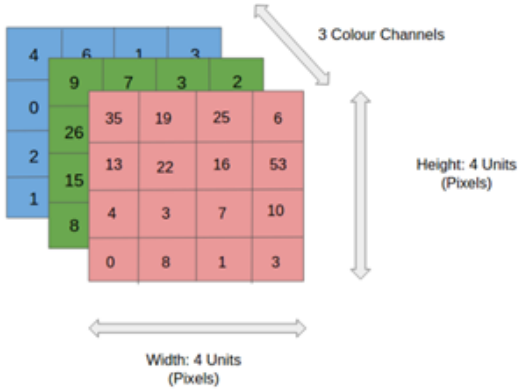
Similarly, for our recurrent neural network, a balanced dataset was create using segments and labels. The segments and labels were then split into our training and testing sets with a test size of 20%. An RNN with Long Short-Term Memory (LSTM) structure was used. This model has three layers, 1 input layer and 2 hidden layers. This LSTM uses 128 memory units (smart neurons). 0.5% of neurons are dropped out in the hidden layer. Two dense layers are added, with activation 'relu' and activation 'softmax'. Because it is a categorical classification problem, categorical crossentropy is used as the loss function. The efficient ADAM optimization algorithm is used. We then proceeded to run 15 epochs, with batch sizes of 80 to feed our model training data. Lastly, a confusion matrix is plotted with evaluation results.

3.2 Convolutional Neural Network

Over the past few decades, deep learning, a potent tool, has been used to handle large amounts of data. In the 1950s, the early years of AI, computer scientists have been trying to build computers to analyse visual data. This study later became to be known as computer vision. In 2012, a group of researchers led by Alex Kritzhevsky made advancements in computer vision by creating an AI system called AlexNet, which used a convolutional neural network. AlexNet won the 2012 ImageNet computer vision contest with an accuracy of 85%. CNN's have thus become pivotal in recent years.

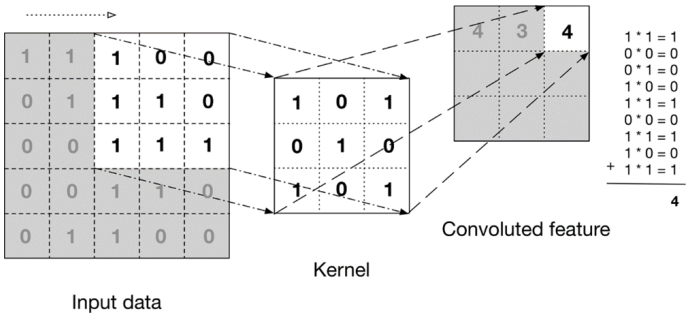
Convolutional neural networks or ConvNets were first introduced in the 1980s by Yann LeCun. ConvNets were used for banking and postal services but could not work efficiently for large images since they needed many data and compute resources.

To understand CNN's, we must first understand what an image is and how we represent it. An RGB image is a matrix of pixel values with three planes representing the primary colours: Red, Green, Blue. A greyscale image is similar but with only one plane and black to white intensity values.



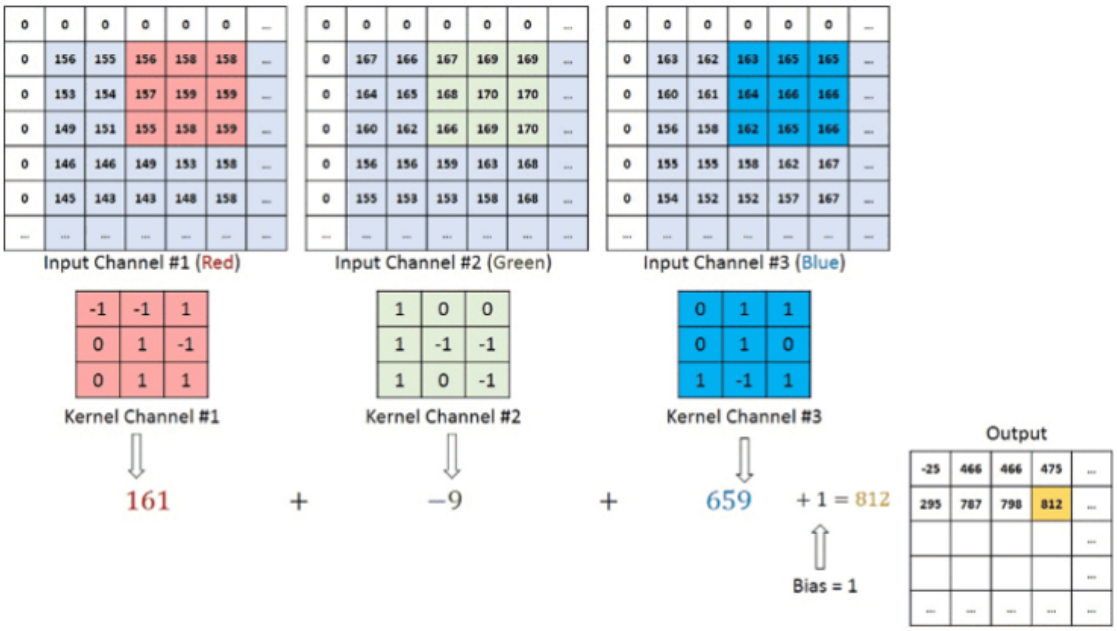
S. Saha, towards data science, Dec. 2018

We will use a greyscale image with black (0) and white (1) pixel values to explain how CNNs work.



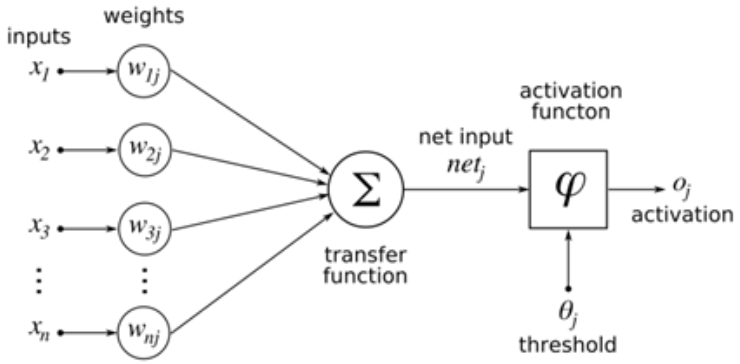
M. Manav, Analytics Vidhya, May 2021.

Above is a demonstrated Convolution. We take a kernel/filter (3x3 matrix) and apply it to the input data, using multiplication to obtain a convolved feature. We pass the convolved feature onto the next layer. For RGB, we do something similar.



S. Saha, towards data science, Dec. 2018

Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons are mathematical functions that calculate the weighted sum of multiple inputs and then output an activation value.



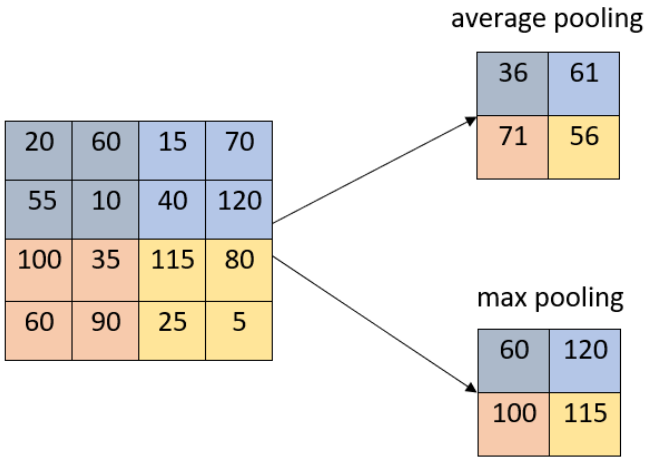
B. Dickson, TechTalks, Jan. 2020.

When a ConvNet takes in an image, each layer generates several activation functions that pass onto the next layer. The first layer usually detects features such as vertical, horizontal, and diagonal edges. The output then passes onto the next layer, which detects more complex features, including corners or combinational edges. Even more complex features such as objects, faces, and other body parts are detected when we move deeper into the CNN.

When interpreting the activation map of the final convolution layer, we see that the classification layer outputs confidence scores in the range $0 - 1$, specifying how likely an image belongs to a class. For example, a ConvNet that can detect frogs, mice, and birds, will output a final layer detailing the possibility that an input image contains any of those creatures mentioned.

Other than the convolutional layer, we also have a pooling layer that reduces the spatial size of the convolved feature. We do this to decrease the computational power required to process the data by reducing the dimensions. We split pooling into two types: Average pooling and max pooling. We will have a look at max pooling.

In max pooling, we find the maximum value of a pixel from a portion of the image covered by the filter. Max pooling also performs as a noise suppressant. The CNN discards noisy activations altogether, and it performs denoising along with dimensionality reduction. However, average pooling returns the average of all values from the portion of the image covered by the filter. Average pooling performs dimensionality reduction as a noise suppressing mechanism; therefore, max pooling is better.



We have decided to use CNNs in our work because of their high accuracy in image classification or 3-dimensional data classification. They are suitable for distinguishing between human actions. [12], [13]

3.3 Limitations of CNN's

A CNN provides in-depth results by recognizing patterns and details that are minuscule and unclear to the human eye, but CNN's, when understanding the contents of an image, perform poorly.

Comfort Work Boots, 2021.

Looking at the above image, a ConvNet will tell a spectator that there are people, food, chairs, a tent, a guitar, and trees in the image. However, it cannot tell a spectator the weather conditions, the location of the camping site, the families ethnicity; whether they are having lunch, breakfast, or supper; their ages; how long they have been camping, and other contents of the image.

If CNNs are used to moderate media content, they may struggle to predict objects despite the amount of training they have. For instance, a sculpture of a naked body might be seen as nude content and will therefore be blocked accidentally.

CNN's also fail to detect objects under different lighting variations or from new angles that they have not seen before. CNN's do not develop the mental models that humans have about different objects.

Another weakness of CNNs is their inability to understand the relations between different objects. The 'Bongard problem,' which introduces two sets of six images, requires distinguishing the critical difference between the sets. For example, there may only be one shape in each image in the first set and two shapes in each image in the second set. If a ConvNet receives a new image to decide which set it belongs to, it will struggle to match the set with the image.

Because a ConvNet is exemplary in examining an image, it becomes vulnerable to adversarial attacks; perturbations in input data will go unnoticed to the human eye and affect a neural network's behaviour. Self-driving cars face these hardships due to their environment and exposure.

Despite these limitations, there is no doubt that convolutional neural networks have caused a revolution in artificial intelligence. As advances in CNN show, our achievements are remarkable and efficient, but we are still far from reproducing the critical components of human intelligence.

3.4 Recurrent Neural Networks

An RNN is a type of deep learning artificial neural network which uses time-series data or sequential data. Deep learning algorithms are commonly used for ordinal or temporal problems. Like feedforward and CNNs, RNNs utilize training data to learn. They are different because of their "memory" as they take information from prior inputs to influence the current input and output. RNN outputs depend on prior elements within the sequence, as opposed to traditional neural networks. RNNs that are bidirectional will depend on future events as well to determine outputs, unlike uni-directional RNNs.

An RNN can be seen as rolled or unrolled. When it is rolled, it represents the whole neural network. If it is unrolled, we have a visual representation of the individual layers or time steps. Another distinguishing characteristic of RNNs is that they share parameters across each layer of the network. RNNs share the same weight parameter within each layer of the network. However, the weights are adjusted through the process of backpropagation and gradient descent to facilitate reinforcement learning.

RNNs leverage backpropagation through time (BPTT) algorithm to establish the gradients; this is different to traditional backpropagation because it is specific to sequence data. BPTT principles are the same as traditional backpropagation, where the model trains itself by calculating errors from its output layer to its input layer. We can fit

the parameters of the model appropriately by adjusting and fitting them. BPTT differs from the traditional approach because it sums errors at each time step, whereas feedforward networks do not sum errors because they do not share parameters across each layer. [14], [15]

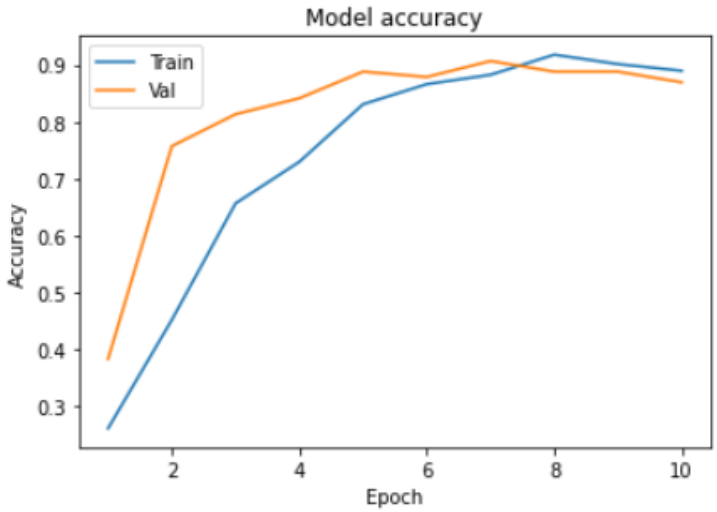
3.5 Long Short-Term Memory

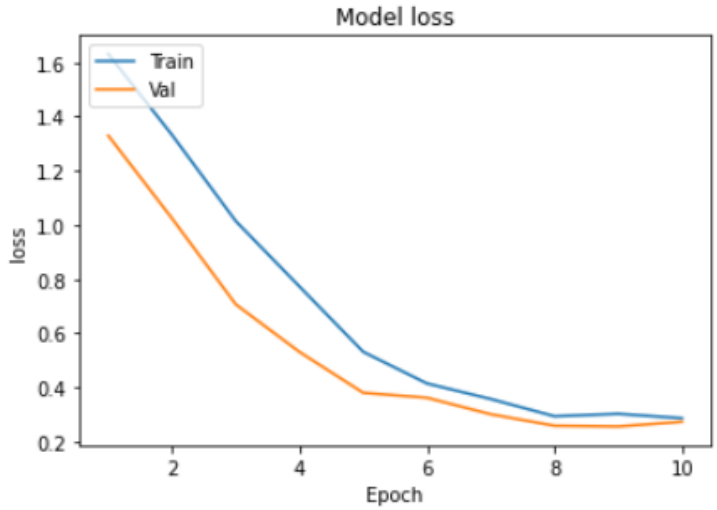
This RNN architecture addresses the problem of long-term dependencies. If the previous state that is influencing the current prediction is not in the recent past, the RNN model may not accurately predict the current state. For example, let us assume that we want to predict the italicized words, "Alice is allergic to nuts. She cannot eat peanut butter." In this context, we can predict Alice doesn't eat nuts, but it would be difficult for the RNN to do this task if the context occurred prior. To fix this, LSTMs have "cells" in the hidden layers of the neural network, which have three gates–an output gate, an input gate, and a forget gate. These gates control the flow of information to predict the output of the network. For example, if the pronoun 'she' is repeated multiple times, you can exclude it from the cell state. [14]

4 Results and Discussion

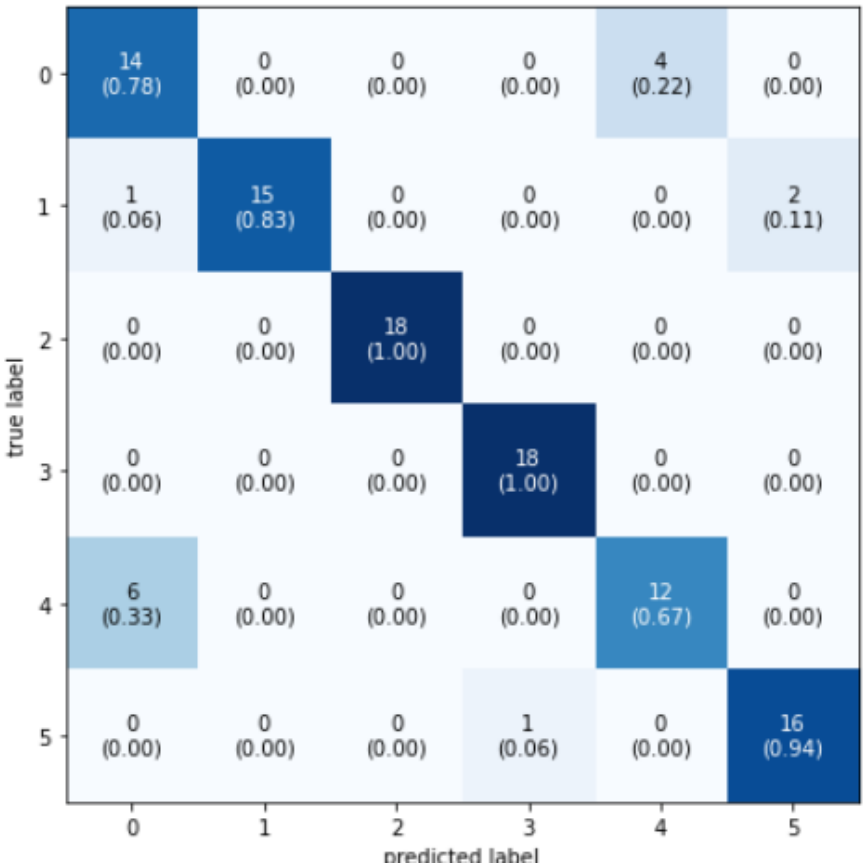
To begin our analysis on the acceleration dataset we have acquired, it is important to recognize the shape of the accelerometer data in all tri-axial directions. This view was plotted in our experiment. These plots showed us that sitting and standing, which are static postures, exhibited low gravitational acceleration and dynamic movements like jogging, upstairs and downstairs have high gravitational acceleration. Movements like upstairs and downstairs are more complicated because they combine vertical, horizontal and especially diagonal movements to a great extent, hence why gravitation has a higher effect on them. Notice that the z-acceleration on motions such as walking and jogging are great. However, jogging has greater acceleration than walking, of course.

Our models are suited for supervised machine learning, which is evident in our dataset. All features (x, y and z) have class labels. As mentioned in our methods, we now split the data into training and testing sets. 20% of the data will be for validation. After that, we feed the data into a 2-dimensional CNN as mentioned in our method and run ten epochs to get better accuracy results on our model. Our CNN model contains 4 layers. We then proceed to plot line graphs for our model's accuracy and model's loss. We obtain the following results.





By running ten epochs on our model, we get a training accuracy of 88.94% and a validation accuracy of 86.92%. We see from the above information that our model is underfitting because the validation loss continues to decrease as we run epochs. This model has a high bias and a low variance. Our model seems to be too simple. This result is good because we are using a simple classifier with smart features. Our validation accuracy is slightly lower than our training accuracy. This result is good because it shows that our model categorises training data with its classes more efficiently. We will now plot a confusion matrix to see how the model handles unseen testing data.



Key:
0 - Downstairs
1 - Jogging
2 - Sitting
3 - Standing
4 - Upstairs
5 - Walking

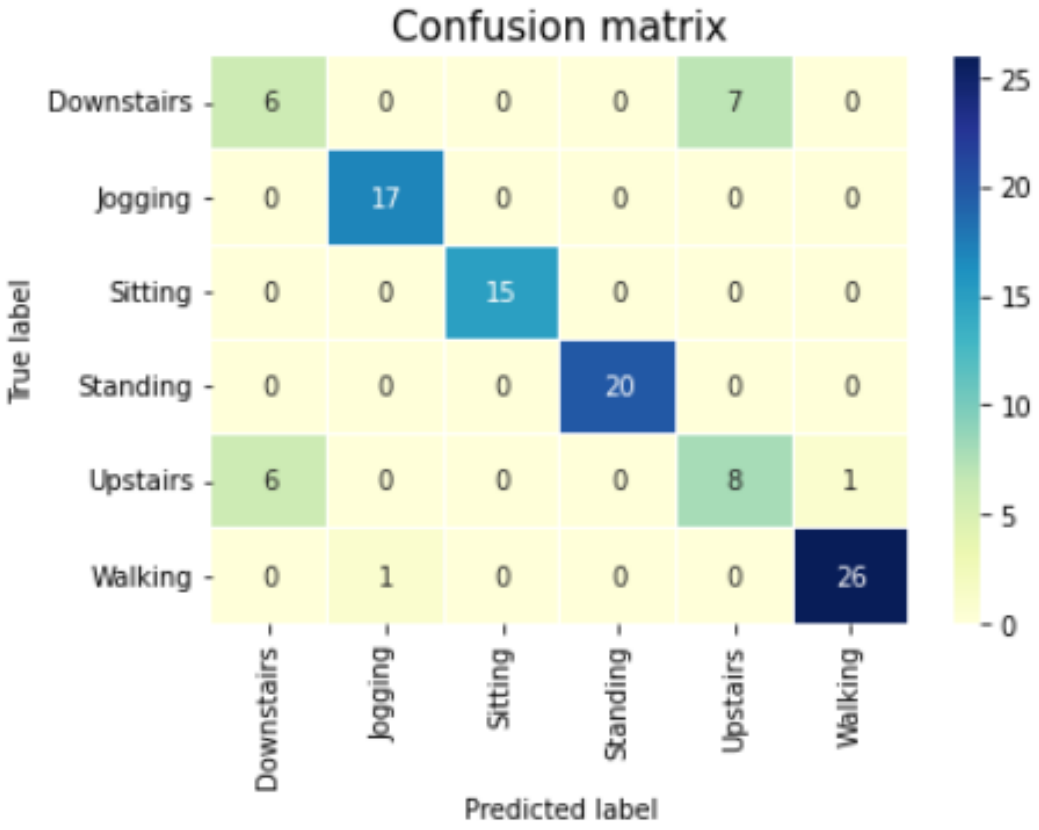
From the above information, we see that the parameters 'Sitting' and 'Standing' are classified with 100% accuracy for our model. Notice that the true label for 'Upstairs' and the predicted label for 'Downstairs' are misclassified with an error of 33%. Vice versa, the true label for 'Downstairs' and the predicted label for 'Upstairs' are misclassified with an error of 22%. This misclassification could be because 'Upstairs' and 'Downstairs' have similar motions. 'Walking' and 'Jogging' have accuracy predictions greater than 80%, so they are classified well. A classification report is compiled, and the accuracy score of our confusion matrix is determined.

We achieved an accuracy score of 87%. This high score shows that our model predicts true classes for our observed values well. Downstairs has a recall of 78%, and Upstairs has a recall of 67%. These values show that the observed cases were not correctly predicted positive well enough for these two attributes. When it comes to

precision, our model is predicting classes correctly really well, except for 'Downstairs', which has an okay precision of 67%. For this model, we predicted 3 unseen test data. We chose 1000 samples from sitting, jogging, and upstairs.

In our analysis, we saw that sitting had a perfect classification score of 100%. This result was accurate for our model since it classifies sitting and standing with 100% accuracy. However, jogging is misclassified drastically with upstairs, achieving an accuracy of 47.83%. This result does not agree with our model, where 83% of jogging was positively classified. For our final test, we used upstairs. Upstairs achieved a positively predicted accuracy score of 0%. In our test, the model predicted upstairs as Standing. These last two results show that our model lacks training data; although the learning model may be smart, it still requires more training data and experience to make better predictions.

We will now investigate the use of a Recurrent Neural Network. Our RNN uses Long Short-Term Memory (LSTM) architecture. In this model, 3 layers were used. 15 epochs are run on the model. Like in our CNN model, the number of epochs run on our RNN model showed that this model was underfitting. The model was evaluated to determine its accuracy and loss. An accuracy of 86% and a loss of 30% was calculated. These calculations are promising. Finally, a confusion matrix was plotted.



From the above confusion matrix, Jogging, Sitting, and Standing are the only two classes that have been classified with 100% accuracy. Walking is misclassified once as Jogging with a 96% accuracy. Downstairs is misclassified as Upstairs in some instances with an accuracy of 46%. Upstairs is misclassified as Downstairs and Walking with an accuracy of 53%. Our models for this research show that Upstairs and Downstairs are similar in their movement since they confuse each other in predictions. In our investigation, RNN's classified jogging and walking better than CNN's. From our study, it can be seen that using an RNN reveals that our data has lots of noise (badly accumulated data) because our RNN performs worse than our CNN.

Compared to recent studies, our accuracy readings are lower. This problem arises because we lack training data, and our dataset does not state the subjects' status' (e.g., height, weight, age and gender) of obtaining the data either or where the smartphone was located. We can assume that it could have been placed anywhere on the body. In work, [5], [6] it is clear that placing an accelerometer near the hip or thigh is most effective as your accuracies will not fall shorter than 90%. We can also see that using just one accelerometer is not as effective as using two, like in work [5].

Our smartphone dataset did not include the use of gyroscopes like in [6], if we used a dataset that was recorded using a gyroscope and accelerometer, we might have obtained better results. Our work only considered 1 dataset for testing as opposed to [6], which used 10 datasets. Using 10 datasets would have been excessive work, but it would have been better to get a more in-depth comparison between CNN and RNN. [6] used RF's; if we considered an ensemble method combining our two networks, we might have obtained better results. Our work uses more subjects as opposed to [6] which is limited to 10, which is better for us.

In work [5], [7], [8] we see that HMM, SVM and DNN can be better at classifying data. A DNN [7] is suited for semi-supervised learning. Our NN is suited for our task and may only be second best to SVM or HMM. In work [7], deep networks are better than shallow networks. This statement is true because our CNN, which had 4 layers, performed better than our RNN, which had 3 layers. The idea of composite activities introduced in [8] would have been interesting to explore; we did not do this. Our classifiers did better than the Observable Markov Model (OMM) classifier used in [8] because they used a single classifier for a single subject, whereas we used more. Our model also performed better because we used tri-axial accelerometers, which gave more accurate results than bi-axial accelerometers. If we compare the activities we used to others used in different related works, it can be seen that static postures are simpler to classify correctly, whereas motion is difficult, especially when ascending and descending stairs.

5 Conclusions and Future Work

We used a machine learning classifier to predict activities based on accelerometer data from smartphones. Our activity list included downstairs, jogging, sitting, standing, upstairs and walking. We used a 2-dimensional convolutional neural network and a recurrent neural network as our classifiers. Our models are said to be supervised learning approaches. Our models perform better with a balanced dataset, which shows learning capacity and produces significant recognition accuracy and better results, versus an unbalanced dataset. Because our models have less training data, their scope is not broad enough to classify unseen test data well. For our CNN model we struggled with classification in Jogging, Downstairs and Upstairs. For our RNN model we struggled with classification in Downstairs, Upstairs and Walking.

Our accuracy scores for both models were very similar. Our CNN model achieved an accuracy of 87%, whereas our RNN model achieved an accuracy of 86%, 1 lower than our CNN model. Our CNN model uses filters and pooling layers, whereas our RNN model feeds results back into the network. From our experiment, it seems using a CNN made more of a bigger difference. A CNN is good for spatial data, and an RNN is more suitable for analyzing temporal and sequential data. The main difference, however, was the number of layers used. Our CNN had more layers than our RNN. Our models have layers that are constructed differently, but the outputs are similar. Our RNN, which should have performed better, also performed worse because it is clear that we have noisy data to learn. Both models performed well, but it seems that activities like Upstairs and Downstairs, which have similar gravitational movements, have made it difficult for our models to obtain even higher accuracies. This struggle can be seen as the downfall of our models. They struggle to classify movement that is similar in shape but different in direction.

For our experiment, it was unclear whether users had to perform their activities with their smartphones in their pockets at the waist level or not. Scientists have proven that accelerometer data captured at the waist level is far more accurate than any other position in the body. However, we can better perfect our model if we use more than one accelerometer sensor simultaneously on different body parts, or if we include other sensor technology, like gyroscopes or RGB data capturers, maybe even sensors that capture skeleton data or depth frames could be useful. These ideas can be researched in future works. In our research our models were reliable at capturing motion that was different in shape, but if the shapes were similar and the direction was different it became a problem. This is why the use of additional sensor technology can be necessary. There are also many means of capturing motion, like smartwatches and smart homes, which can be investigated in future works. With the increase in investigations of artificial intelligence, human activity recognition is making breakthroughs in the technology sector aiming at the welfare of humanity.

References

[1] N. Qamar, N. Siddiqui, M. Ehatisham-ul-Haq, M. A. Azam, and U. Naeem, "An approach towards position-independent human activity recognition model based on wearable accelerometer sensor," *Procedia Computer Science*, vol. 177, pp. 196–203, 2020.

[2] M. Ehatisham-ul-Haq, M. A. Azam, Y. Asim, Y. Amin, U. Naeem, and A. Khalid, "Using smartphone accelerometer for human physical activity and context recognition in-the-wild," *Procedia Computer Science*, vol. 177, pp. 24–31, 2020.

[3] A. Bayat, M. Pomplun, and D. A. Tran, "A study on human activity recognition using accelerometer data from smartphones," *Procedia Computer Science*, vol. 34, pp. 450–457, 2014.

[4] S. R. Shakya, C. Zhang, and Z. Zhou, "Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data," *Int. J. Mach. Learn. Comput*, vol. 8, no. 6, pp. 577–582, 2018.

[5] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay, "Optimal placement of accelerometers for the detection of everyday activities," *Sensors*, vol. 13, no. 7, pp. 9183–9200, 2013.

[6] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," *Ieee Access*, vol. 5, pp. 3095–3110, 2017.

[7] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[8] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[9] J. W. Lockhart, T. Pulickal, and G. M. Weiss, "Applications of mobile activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1054–1058.

[10] G. M. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[11] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[13] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375.

[14] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.

[15] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.