Human Activity Recognition using Convolutional Neural Network

Muzerengwa Vincent, MZRVIN001

Abstract

This is report about a Convolutional Neural Network (CNN)-based Human Activity Recognition (HAR) system using smartphone sensor data from the MotionSense dataset. The model processes accelerometer and attitude signals through a 1D CNN architecture, achieving 97% test accuracy in classifying six activities: walking, jogging, sitting, standing, and ascending/descending stairs. Key pre-processing steps include sliding window segmentation and feature normalization. Results show near-perfect recognition for static activities (sitting, standing) and jogging, while stair related motions had greater challenges in classification. The report shows that CNNs can effectively extract spatal and temporal features for HAR, with potential applications in health monitoring, fitness tracking, and smart environments.

I. Introduction

HUMAN activity Recognition (HAR) is a field of study that aims to classify physical activities performed by humans through the use of sensor data. It is used in various fields such as healthcare, sports analytics, smart homes, fitness tracking and assisted living systems. HAR system enables real-time monitoring and automated recognition of human behavior, providing useful information that could be used to improve user experience in these applications.

Inertial Measurement Units(IMUs) can be embedded onto wearable devices such as smartphones. This has made it easier to collect motion data such as acceleration, gyroscope and orientation. The raw data from the sensors is usually high-dimensional and temporally dependent, which makes time-series analysis and signal processing neccessary components for HAR development.

This report will use MotionSense dataset, a public dataset collected using iPhone's embedded sensors while participants performed different activities. The goal is to develop a machine learning classifier that can accurately classify the different activities participants are performing based on sensor data.

The aim is to learn spatial and temporal features from the raw and pre-processed sensor data using a Convolutional Neural Network(CNN). The CNN is trained and performance measured using metrics such as classification accuracy and confusion matrix analysis. The report will outline the design process, implementation steps, evaluation of results and analysis of performance.

II. Literature Review

Human Activity Recognition(HAR) is a highly researched field. This is due to its broad application in areas such as healthcare, smart homes and fitness tracking.

A. Privacy concerns in using time-series data.

Studies have investigated techniques for optimizing privacy and performance of HAR using mobile and wearable sensors. This is because sensor data may reveal other activities, such as weight, or smoking habits, that a user might want to remain private.

Malekzadeh [1] investigated mobile sensor data anonymization with the main focus being the privacy risk associated with sharing time-series motion data. It was emphasized that there was need to carefully select features and transformations that minimize the risk of identity leakage while retaining task-relevant information.Because of these concerns, it is important to perform appropriate pre-processing on the raw data, such as using magnitudes or derived features, in privacy-sensitive applications.

B. Data Pre-Processing Techniques

A study comparing various data preprocessing approaches for deep learning-supported HAR tasks found that spectrogram representations of acceleration data can enhance the extraction of interpretable features, leading to improved classification accuracy[2]. The effectiveness of such transformations depends on the dataset's characteristics and the specific activities being recognized. Although spectrogram-based transformations have shown promise in enhancing feature extraction in HAR tasks, this report will use raw time-series data with a 1D CNN due to its lower computational overhead and effectiveness in capturing temporal dependencies in sensor readings.

C. Model Architecture

Ahmad [3] proposed a method combining multi-head Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks to recognize human physical activities. This architecture effectively captures both spatial and temporal features from sensor data, leading to improved recognition accuracy. However, this method is complex to implement and has some serious resource constraints. Future models could incorporate these hybrid models for deeper feature learning.

In a follow-up work, Malekzadeh [4] proposed a dimension-adaptive neural architecture(DANA) made for multivariate time-series sensor data. The study shows that adaptive architectures can achieve high performance while maintaining compact model sizes, which is crucial for on-device deployment. This supports the use of convolutional neural networks (CNNs) for this report, which are capable of learning spatially and temporally relevant features without requiring complex models.A single-branch 1D CNN architecture will balance complexibility and resource constraints.

III. Method

This section will look at the pipeline developed for Human Recognition(HAR) using MotionSense dataset. The methodology will cover data acquisition, pre-processing steps, windowing, feature normalization, label encoding, data splitting and model architecture.

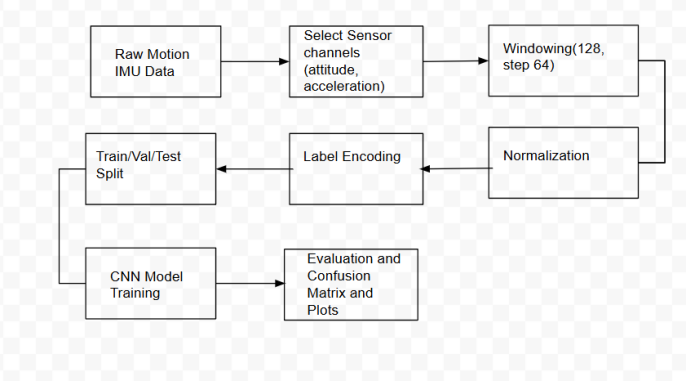

Fig. 1. Flowchart of model pipeline

A. Data Acquisition

The MotionSense dataset provides data recorded from iPhones at a frequency of 50Hz. Each recording corresponds to a subject performing one of six activities: walking(wlk), jogging(jog), walking upstairs(ups), standing(std), and sitting(sit). Sensor readings include:

- Attitude: roll, pitch, yaw
- User Acceleration: z, y, z axes
- Gravity: z, y, z axes

A custom data loading function constructs a labeled dataset by:

- Select specific sensor types
- Computing the raw 3D values (not magnitudes)
- Combining readings with metadata (activity, subject ID, age, gender, etc.)

For this report, only user acceleration and attitude will be used. These two sensors capture both dynamic movement and orientation, which are crucial for recognizing physical activities.

1) Reducing input features: Omission of Gravity: Gravity was excluded from the dataset for this experiment. This is because:

- Gravity is relatively constant or slowly changing over time. Therefore, it will have little impact on the results.
- It can be inferred from acceleration and attitude data.
- It adds redundancy or low-informitive features to the task at hand.

2) Reducing input features: Omission of Rotation Rate: Gyroscope was not used in order to simplify the model. This is because it introduces more sensor noise and the target classes are well characterized by acceleration and orientation.

B. Data pre-processing

1) Sensor Selection: From the previous section, it has already been established that two sensors will be used for this experiment:

- attitude (roll, pitch, yaw)
- userAcceleration (x, y, z)

This results in six input channels. The selected data will provide directional information relevant for distinguishing similar activities.

2) Windowing: Sensor data is a continuous stream of readings over time. The sliding window technique will be used to slide this continuous stream into fixed-length segments, which will be used as training samples for the sample. Each sample will preserve the temporal structure of the activity within that segment. Using overlapping windows increases the number of training samples without need for more data collection. More samples will result in better model generalization and improved model accuracy. A window size of 128 samples of 128 samples will be used with 64 samples will be used to achieve a 50% overlap.

3) Label Encoding: Activity labels are mapped to integer classes using label encoder for compatibility with categorical cross-entropy loss function. Label encoding will transform categorical activity labels into numerical format that can be used by neural networks.

4) Normalization: Raw data is collected from different sensors. Each feature is on different numeric scale, which could be a problem for the neural network. Therefore, each windowed segment is normalized across each feature dimension to standardize sensor input ranges. This helps speed up training and stabilizes convergence. The normalization equation is shown below:

$$X_{\text{norm}} = \frac{X - \mu}{\sigma} \tag{1}$$

If not normalized, features with larger scales dominate the learning process, causing the model to learn suboptimal weights.

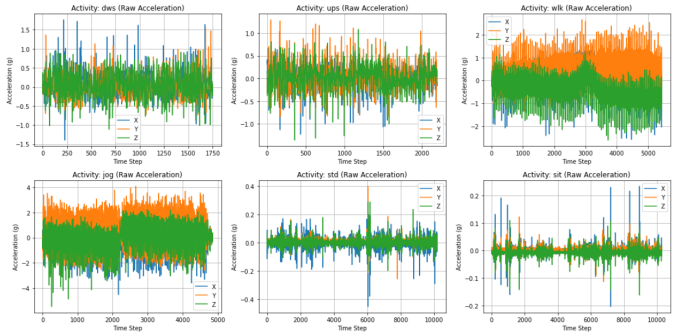

Fig. 2. Sample of activities data from Acceleration sensor before pre-processing(raw data)

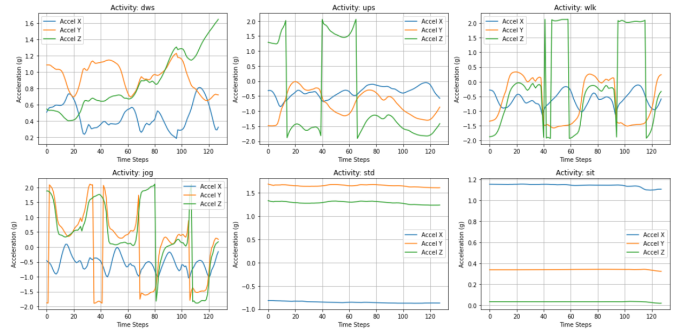

Fig. 3. Sample of activities data from Acceleration sensor after pre-processing

C. Resampling

For this report, resampling was not required because all the time-series data collected from the MotionSense dataset was already uniformly sampled. Each sensor stream was recorded at a consistent and fixed sampling frequency throughout the trials. This uniformity in temporal resolution ensured that each time-series segment had regular intervals between data points, making it unnecessary to perform any resampling operations to align or correct timing inconsistencies.

D. Model Design

The model is made up of a series of convolutional and pooling layers that automatically extract hierarchical temporal features from the normalized input sequences. The first convolutional layer employs 64 filters with a kernel size of 3, followed by a max pooling layer that downsamples the feature maps. A second convolutional layer with 128 filters further refines the learned features, and another pooling layer reduces dimensionality. The output of the convolutional blocks is flattened into a one-dimensional vector, passed through a dropout layer to avoid overfitting and then processed by fully connected dense layers. The final dense layer uses a softmax activation function to output probability distributions over the six activity classes.

TABLE I
CONFIGURATION OF 1D CNN FOR HAR USING MOTIONSENSE

Layers	**Parameter**
Input	Shape = (128 $\times$ 6)
Convolution 1D	Filters = 64, Kernel size = 3, Stride = 1, Activation = ReLU
Max Pooling 1D	Pool size = 2, Stride = 2
Convolution 1D	Filters = 128, Kernel size = 3, Stride = 1, Activation = ReLU
Max Pooling 1D	Pool size = 2, Stride = 2
Dropout	Dropout rate = 0.5
Flatten	-
Fully Connected (Dense)	Units = 100, Activation = ReLU
Output Layer (Dense)	Units = 6 (classes), Activation = Softmax

The first CNN layer (64 filters) captures low-level, local features like directional movement patterns from the accelerometer signal. The second CNN layer (128 filters) learns higher-level, more abstract features, such as complex activity patterns, for example walking vs. jogging, by combining and building on the earlier features. Too many filters can overfit and increase computational cost, especially for smaller datasets such as this one.

After flattening the time series features from convolutional layers, the dense layer has a large number of parameters. Applying dropout at 0.5 before the final classification layer helps reduce the risk of overfitting to training data. A dropout rate of 0.5 has been widely used and shown to be effective in fully connected layers of neural networks [5].

A kernel size of 3 means each filter looks at 3 consecutive time steps at a time. This is ideal for capturing local temporal patterns, like quick changes in motion or transitions between activities.Small kernel sizes are faster to compute and require fewer parameters, reducing the chance of overfitting.

A max pooling layer with a pool size of 2 is used after each convolutional layer. This configuration means that the model takes the maximum value every two consecutive time steps in the feature maps. By default, the stride is set to 2, resulting in non-overlapping pooling windows. This approach reduces the temporal resolution by half at each stage while keeping features of the signal, reducing computational complexity.

E. Training Setup

The dataset was partitioned using a 3-way data split strategy, 70% of the data for training, 15% for validation, and 15% for testing. This split ensured the distribution of activity classes across all subsets which allowed the model to learn and make good generalizations.

For the loss function, categorical cross-entropy was used, as it is well-suited for multi-class classification problems where the output is a probability distribution over several classes. The optimizer selected for training was Adam.This is because it is effective in handling noisy gradients and sparse data.

The model was trained for 15 epochs with a batch size of 64. The number of epochs was sufficient to allow the model to learn meaningful patterns in the data without overfitting, as monitored by performance on the validation set. The chosen batch size ensured efficient use of memory and accelerated computation during training.

IV. RESULTS

This section shows the performance of the CNN-based Human Activity Recognition (HAR) model trained on the MotionSense dataset. After applying all preprocessing steps: sensor selection, windowing, label encoding, and normalization, the data was

split into training, validation, and test sets to evaluate the model's effectiveness in classifying the six distinct human activities: walking, jogging, sitting, standing, going upstairs, and going downstairs.
The final trained model achieved a test accuracy of 91.8%. This means that the model could strongly distinguish between the activities based on accelerometer and attitude sensor data.

A. Classification Report and F1 scores

TABLE II
CLASSIFICATION PERFORMANCE METRICS

Class	Precision	Recall	F1-Score	Support
dws	0.94	0.88	0.91	301
ups	0.89	0.93	0.91	362
wlk	0.99	0.97	0.98	800
jog	0.99	1.00	0.99	307
std	0.98	1.00	0.99	711
sit	1.00	1.00	1.00	788
accuracy		0.97		3269
macro avg	0.96	0.96	0.96	3269
weighted avg	0.97	0.97	0.97	3269

dws (Downstairs Walking)
Precision is 0.94, meaning 94% of predicted "dws" samples were correct. Recall is 0.88, suggesting 12% of actual "dws" samples were misclassified. The F1-score of 0.91 indicates good performance. The other 0.09 may be because some downward walking instances may resemble other movements (e.g., walking or stairs up).

ups (Upstairs Walking)
Precision is 0.89 and recall is 0.93. This suggests the model correctly identifies most "ups" samples but sometimes mistakes other classes for it. The F1-score is 0.91, similar to dws.

wlk (Walking)
With a precision of 0.99 and recall of 0.97, walking is very well classified, with few false positives or negatives. The F1-score of 0.98 indicates excellent performance.

jog (Jogging)
Near-perfect performance: precision of 0.99, recall of 1.00. The model rarely misclassifies jogging and correctly identifies all instances. F1-score is 0.99.

std (Standing)
The model achieves very high scores: precision of 0.98 and perfect recall of 1.00, meaning no standing sample was missed. This is confirmed by an F1-score of 0.99.

sit (Sitting)
Perfect classification with scores of 1.00 for precision, recall, and F1. The model distinguishes sitting extremely well, likely due to its clear sensor pattern.

Overall accuracy
Out of the 3269, only 3% were misclassified. This shows that the model has a high level of accuracy. The macro average 0f 0.96 shows the unweighted average of scores across all classes, treating each activity equally. It shows the model is consistently accurate across classes, not just biased toward frequent ones. The weighted average of 0.97 accounts for class imbalance by weighting each class by its number of instances. The higher value here confirms that the model performs especially well on more frequent activities.

B. Confusion Matrix

The confusion matrix provides a detailed view of the model's classification performance across all six activities: walking, jogging, sitting, standing, upstairs, and downstairs. An ideal classifier will have all predictions concentrated along the diagonal of the matrix, indicating that the predicted labels match the true labels exactly. In this case, the confusion matrix is almost entirely diagonal. This confirms that the CNN model consistently and accurately predicted the correct activity class for most of the test instances.
The minor off-diagonal entries suggest a few misclassifications, possibly due to overlapping sensor patterns (e.g. subject walking from upstairs to downstairs). However, these do not significantly impact the overall performance.

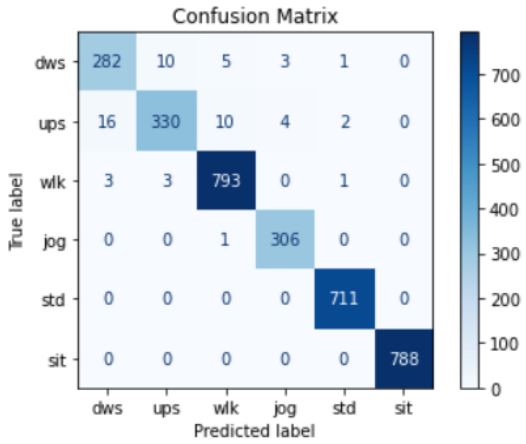

Fig. 4. Confusion Matrix

C. Training and Validation Accuracy

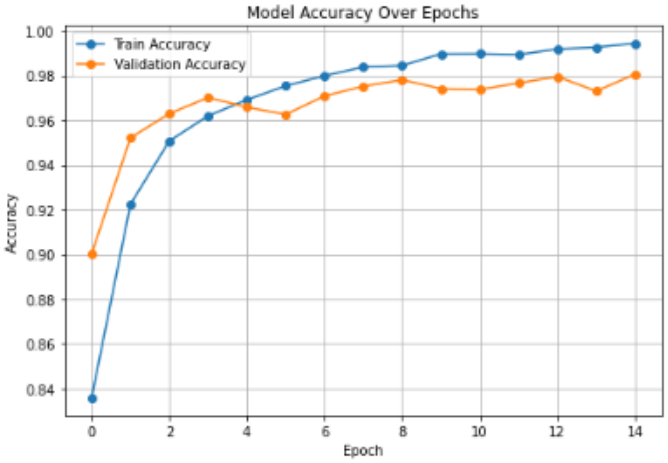

Fig. 5. Model accuracy over Epoch

The training accuracy begins around 83% and rises quickly, reaching approximately 99% by the final epoch. The validation accuracy also shows a strong upward trend, stabilizing around 98%.This shows that the model generalizes well to unseen data.

The small and consistent gap between training and validation accuracy suggests minimal overfitting. It shows that the model has effectively learned to extract relevant features from the data.

D. Training and Validation Loss

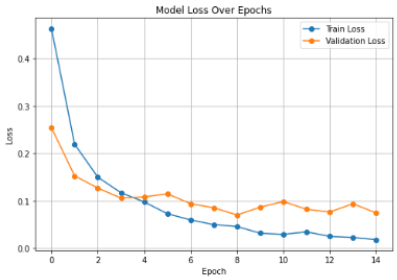

Fig. 6. Model loss over Epoch

(Figure 6) shows training and validation loss decrease rapidly in the first few epochs. Training loss reaches a value below 0.02, while validation loss stabilizes around 0.08 to 0.10 in the later epochs. Although there's a slight fluctuation in validation loss, it remains low and shows no sign of significant overfitting. This confirms that the model maintained a good fit to the validation data throughout training.

V. Analysis of Results

With an overall test accuracy of 97% and macro and weighted F1-scores of 0.96 and 0.97 respectively, the CNN model proved highly effective in distinguishing between the six defined activities using only acceleration and attitude data.

A. Per-class Metrics

Activities such as jogging, sitting, and standing achieved near-perfect or perfect precision and recall, suggesting that their movement patterns are both distinct and consistent across users. These results indicate that the model is capable of capturing strong discriminative features for stationary and rhythmic activities.
However, the model as slightly less accurate for walking upstairs (ups) and walking downstairs (dws). These two classes had lower F1-scores (0.91 each) compared to the other activities. This performance drop is likely due to their similarity to regular walking and possibly each other, as all three involve similar leg movements and acceleration profiles. The vertical motion patterns of ups and dws can vary more across users, depending on stride length, speed, or device positioning, introducing inconsistencies that make classification more challenging.

The perfect classification of the sitting activity is notable, likely due to its distinct and steady accelerometer and attitude signature. This aligns with existing findings in HAR literature, where static postures often produce minimal variance in sensor readings and are thus easier to detect.

B. Training and Validation Curves

The training and validation curves converged. Both accuracy and loss stabilized after around 10–12 epochs, and there was no significant overfitting. Validation accuracy remained close to training accuracy, and validation loss did not increase towards the end of training. These trends show that the model is well-regularized and trained appropriately without memorizing the training data.

C. Comparing 3-way split to k-fold cross-validation

The model achieved high accuracy and F1-scores using a 3-way data split (70% training, 15% validation, 15% test).However, a fixed split could cause potential variability. In real-world scenarios, a model's performance may vary depending on how the data is partitioned, particularly for time-series data or datasets with slight class imbalance.
To solve this problem, k-fold cross-validation can be used. In this method, the dataset is divided into k equal parts (commonly 5 or 10), and the model is trained and validated k times, each time using a different fold for validation and the rest for training. This process ensures that every data point is used for both training and validation, reducing the risk of biased performance estimates. A simulation was done using 5-fold cross validation and produced an F1 score of 0.9777. The high accuracy of the two score reveal that the model's accuracy is consistently replicable

VI. Limitations and Future Improvements

Despite the strong performance, the model could benefit from further improvements. For example, exploring feature extraction methods like spectrogram-based transformations could improve sensitivity to complex motion patterns such as differentiating between ascending and descending stairs. Additionally, implementing data augmentation techniques or experimenting with more advanced architectures such as CNN-LSTM hybrids might help and improve model generalization.

VII. Conclusion

In conclusion, this report successfully developed a human activity recognition (HAR) system using a convolutional neural network (CNN) trained on the MotionSense dataset. The dataset, made up of time-series data from smartphone motion sensors, was preprocessed through feature extraction, windowing, normalization, and one-hot encoding. The system was designed with a clear and structured pipeline that ensured the data was consistent and the model was efficient.

The CNN model achieved a high test accuracy of 97%, with strong performance across all activity classes. The F1-scores for individual activities ranged from 0.91 to 1.00, showing the model's balance of precision and recall. These results show how effective the model is in distinguishing between various physical activities such as walking, jogging, sitting, standing, ascending and descending stairs. Furthermore, training and validation accuracy and loss plots indicated stable convergence with

no significant overfitting.

Overall, the framework developed in this report show that CNNs can be used for HAR tasks if data preprocessing is carefully structured. This framework could be extended to real-world applications in healthcare, fitness tracking, or smart environments, with potential enhancements such as model compression, real-time optimization, or sensor fusion.

REFERENCES

[1] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi, "Mobile sensor data anonymization," in Proceedings of the International Conference on Internet of Things Design and Implementation, ser. IoTDI '19. New York, NY, USA: ACM, 2019, pp. 49–58. [Online]. Available: http://doi.acm.org/10.1145/3302505.3310068

[2] X. Zheng, M. Wang, and J. Ordieres-Meré, "Comparison of Data Preprocessing Approaches for Applying Deep Learning to Human Activity Recognition in the Context of Industry 4.0," Sensors, vol. 18, no. 7, p. 2146, Jul. 2018. [Online] Available at: https://doi.org/10.3390/s18072146

[3] W. Ahmad, M. Kazmi, and H. Ali, "Human Activity Recognition using Multi-Head CNN followed by LSTM," arXiv preprint arXiv:2003.06327, Feb. 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2003.06327

[4] M. Malekzadeh, R. Clegg, A. Cavallaro, and H. Haddadi, "Dana: Dimension-adaptive neural architecture for multivariate sensor data," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 5, no. 3, pp. 1–27, 2021.

[5] Vishnuam. "Dropout in Convolutional Neural Networks (CNN) - Vishnuam - Medium." Medium, 7 Oct. 2024, medium.com/@vishnuam/dropout-in-convolutional-neural-networks-cnn-422a4a17da41. Accessed 29 May 2025.