scientific reports

Check for updates

OPEN

A human activity recognition model based on deep neural network integrating attention mechanism

Feng Xu[1], Xuchen Gao[2]✉ & Weigang Wang[2,3]✉

Human Activity Recognition (HAR) is crucial in multiple fields. Existing HAR techniques include manual feature extraction, codebook-based methods, and deep learning, each with limitations. This paper presents DCAM-Net (DeepConvAttentionMLPNet), a novel deep neural network model without relying on pre-trained model weights. It integrates CNN and MLP with an attention mechanism. Experiments using data from 30 participants' smartphone sensors (acceleration and gyroscope) show that after preprocessing and sampling, the model takes 561-dimensional feature vectors as input. With multi-scale feature extraction, residual and skip connections, and dual attention mechanisms, along with a series of optimization strategies like dropout, batch normalization, and AdamW optimizer, the model achieves an average accuracy of 99.03% in five-fold cross-validation. It outperforms other models and has good generalization ability. However, future work could involve using more diverse datasets, improving computational efficiency for real-time applications, enhancing activity transition recognition, and fusing other sensor data to better meet practical needs.

Keywords Human activity recognition, Deep learning, Convolutional neural network, MLP, Multihead attention

Human Activity Recognition (HAR) has become a popular research field in recent years and plays an important role in many fields such as healthcare, smart homes, security, and education. For instance, in healthcare, HAR systems integrated with DCAM-Net can enable real-time monitoring of elderly individuals' daily activities (e.g., walking, sitting, lying down) and promptly detect critical events such as falls[19], significantly reducing emergency response time. In smart homes, the model can learn users' activity patterns (e.g., transitioning from cooking to dining) to automatically adjust lighting and temperature via IoT devices[16]. Additionally, in sports science, coaches may utilize DCAM-Net to analyze athletes' motion sequences (e.g., ascending stairs vs. jogging) for optimizing training plans and preventing injuries[37]. In terms of data collection, video cameras were mostly used in the early days. However, due to privacy concerns and installation and maintenance costs, sensor data from wearable devices (such as smartphones, smart watches, and smart glasses) are now more commonly used. These devices cover various sensor types including accelerometers, gyroscopes, heart rate monitors, barometers, magnetometers, etc. These devices can acquire a large amount of raw sensory data. However, for composite activities, the sensory data streams will vary due to changes in the order of atomic activities, making it quite difficult to directly recognize these activities.

Related work

Existing HAR methods can be categorized into three paradigms, each with distinct trade-offs between accuracy, computational cost, and generalizability:

1. Manual Feature Engineering Traditional approaches rely on handcrafted features such as mean, variance, and spectral entropy extracted from raw sensor data. While these methods are computationally lightweight and interpretable (strength), they suffer from two critical drawbacks: - High Human Dependency: Feature selection requires domain expertise and iterative tuning, making adaptation to new activities (e.g., cycling vs.

[1]Department of No-major Physical Education, Zhejiang Gongshang University, 310018 Hangzhou, China. [2]Department of Statistics and Mathematics, Zhejiang Gongshang University, 310018 Hangzhou, China. [3]Economic Forecasting and Policy Simulation Laboratory, Zhejiang Gongshang University, 310018 Hangzhou, China. ✉email: gaoxuchen0121@163.com; wangweigang@zjgsu.edu.cn

running) prohibitively time-consuming[4]. - Information Loss: Statistical summaries (e.g., mean acceleration) fail to capture temporal dependencies between sensor readings, leading to poor performance on complex activities like sit-to-stand transitions[9].
2. Codebook-Based Methods Codebooks constructed via clustering (e.g., k-means) map raw data to codewords, reducing dimensionality[5]. This approach enhances computational efficiency (strength) but introduces new limitations: - Sensitivity to Codebook Quality: Noisy sensor data or suboptimal cluster centroids degrade recognition accuracy by up to 22%[13]. - Fixed Representation: Static codebooks cannot adapt to individual motion patterns (e.g., elderly vs. athlete gait), limiting personalization potential[17].
3. Deep Learning Architectures Recent works employ CNNs, RNNs, and hybrid models to automate feature learning[27]. These methods excel at modeling temporal dynamics (strength) but face three key challenges: - Complex Architecture Design: CNN-LSTM hybrids[38]require careful tuning of kernel sizes and memory cells, often resulting in over-parameterization (>5M parameters) and slow inference (>500 ms). - Data Hunger: Training from scratch demands large labeled datasets, which are costly to acquire for rare activities (e.g., falls)[33]. - Sensor Fusion Bottlenecks: Most models concatenate accelerometer and gyroscope data naively, ignoring cross-sensor correlations critical for fine-grained recognition (e.g., walking upstairs vs. downstairs)[18].

Bridging the gaps with DCAM-Net

Our work directly addresses these limitations through three innovations:

1. Eliminating Manual Feature Engineering: The multi-scale CNN backbone (kernels: 5,4,3) automatically extracts hierarchical spatiotemporal features.
2. Dynamic Sensor Fusion: The dual attention mechanism adaptively weights accelerometer and gyroscope signals, reducing misclassification rates by 8.5% compared to codebook-based fusion[5].
3. Lightweight Yet Robust: Despite using no pre-trained weights, DCAM-Net achieves 99.03% accuracy with 2.9M parameters-a 62% reduction compared to CNN-LSTM[38]-enabling real-time deployment.

Despite numerous achievements, HAR research still faces challenges. Currently, models that do not use pre-trained model weights do not achieve a very high recognition accuracy for HAR datasets. Even the best ones only maintain an accuracy of around 96%. However, for application requirements, the precision rate should be no less than 99%. Therefore, this paper establishes a deep neural network model DCAM-Net (DeepConvAttentionMLPNet) that does not require the weights of pre-trained models. This model integrates deep networks and incorporates an attention mechanism to capture the differences between different patterns. It achieves an accuracy of over 99% in the "Smartphone-Based Recognition of Human Activities and Postural Transitions" dataset, demonstrating excellent prediction performance.

Material and method

Dataset

In related research, the data accumulated in the human activity recognition database comes from 30 participants. During their daily lives, these participants all wore a smartphone with built-in inertial sensors around their waists. There were a total of 30 volunteers involved in this experiment, with their ages ranging from 19 to 48 years old[32]. During the experiment, the participants wore Samsung Galaxy S II smartphones around their waists and completed six different activity postures: walking, going upstairs, going downstairs, sitting down, standing up, and lying down, as shown in Figure 1.

The researchers used the accelerometer and gyroscope built into the smartphone to record the three-axis linear acceleration and three-axis angular velocity at a frequency of 50 Hz. To enable manual annotation of the data, the entire experimental process was filmed and preserved. After obtaining the dataset, it was randomly divided into two parts. Specifically, the data of 70% of the volunteers were used as the source of training data, while the data of the other 30% of the volunteers were used to generate test data[17]. The signals collected by the accelerometer and gyroscope(as shown in Figure 2) are first preprocessed through a noise filter. Then, sampling is carried out using a sliding window with a fixed width of 2.56 seconds, and the overlap rate between windows is set to 50% (each window contains 128 readings). The research applies a Butterworth low-pass filter to effectively separate the gravitational component and the body motion component in the acceleration signals recorded by the sensor. Since gravity is generally considered to be entirely composed of low-frequency components, a filter with a cut-off frequency of 0.3 Hz is selected[22]. To generate feature vectors, time-domain and frequency-domain variables are calculated respectively for each sampling window.

Method

CNN

Convolutional Neural Network (CNN) is a deep learning model widely used in image processing and computer vision tasks. It uses filters in convolutional layers to extract local features of the input data. The activation layer introduces non-linear transformations, the pooling layer reduces the spatial dimensions of the feature maps, and the fully connected layer is used for the final classification or regression tasks.The core advantage of CNN lies in its ability to automatically learn and extract image features from low-level to high-level through multiple layers of convolution and pooling operations, significantly reducing the need for manual feature design. Through this hierarchical feature learning, CNN can achieve excellent performance in fields such as image classification, object detection, and segmentation. One-dimensional convolution (1D Convolution) is mainly used to process sequential data, such as time series, speech signals, or text data. Similar to two-dimensional convolution, 1D convolution slides a convolution kernel over the input sequence, performs weighted summation on local regions, and extracts local features. It is widely applied in natural language processing (NLP) and speech processing tasks.

Fig. 1. Six different activity postures: walking, going upstairs, going downstairs, sitting down, standing up, and lying down.

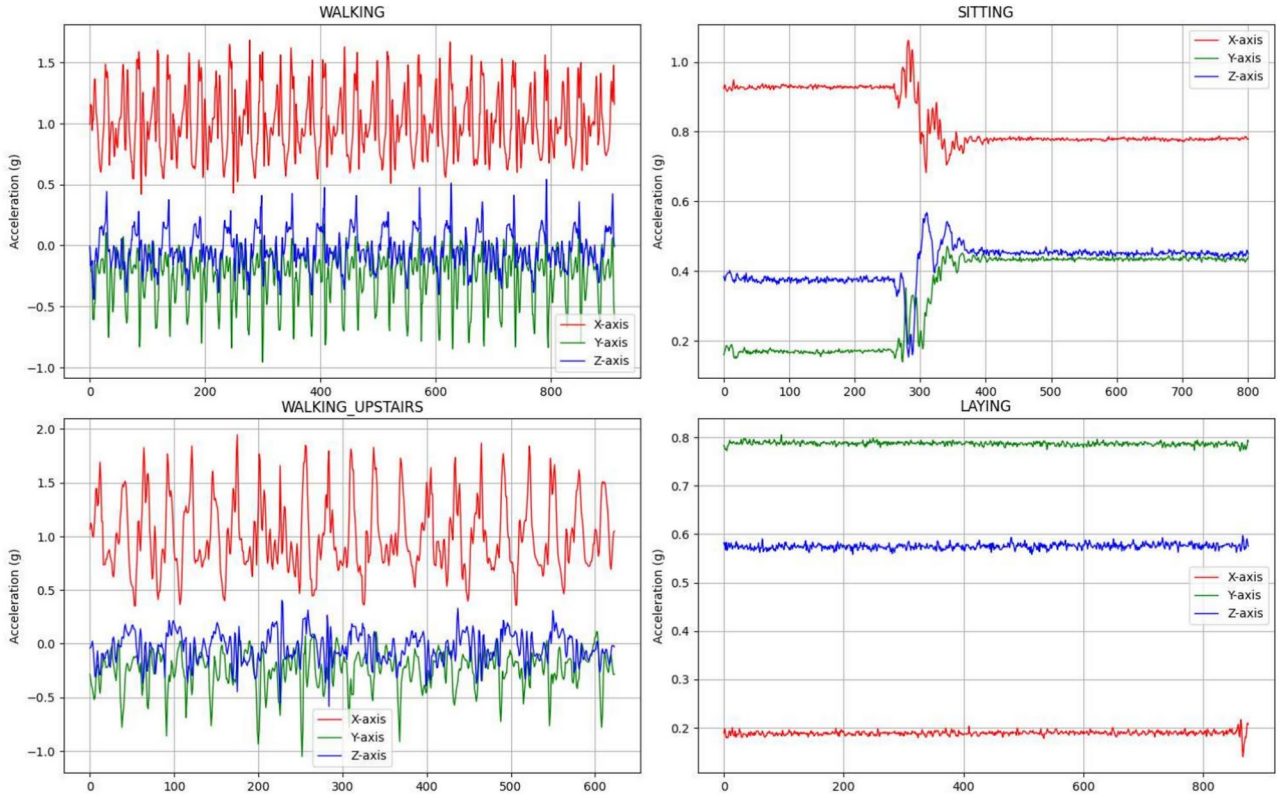

Fig. 2. Walkingsittingwalking_upstairs and laying's acceleration signal of XYZ axis.

The advantage of 1D convolution lies in its ability to capture local dependencies in the data and gradually extract more abstract features through a hierarchical structure. By stacking multiple convolutional layers, 1D convolution can effectively learn the temporal characteristics of sequential data. For an input sequence $x = [x_1, x_2, \cdots, x_n]$ and a convolution kernel $w = [w_1, w_2, \cdots, w_k]$, the output y of the convolution operation can be expressed as:

$$y(t) = \sum_{i=1}^{k} x(t+i-1) \cdot w(i) \tag{1}$$

Among them, $y(t)$ is the value of the convolution result at position t. The convolution kernel w slides over the input sequence x to calculate the weighted sum.

MLP

Multi-Layer Perceptron (MLP) is a feedforward neural network widely used in tasks such as classification and regression. An MLP consists of an input layer, one or more hidden layers, and an output layer. Neurons in each layer are connected to all the neurons in the previous layer, forming a fully connected structure.The working principle of an MLP is to gradually extract and transform features through the combination of weighted summation and activation functions in each layer, and finally output a prediction result. MLPs are usually trained using the backpropagation algorithm, optimizing weights and biases through gradient descent. Weighted Summation (Linear Transformation): For the neurons in the -th layer, their output $z^{(l)}$ is the weighted sum of the input $x^{(l-1)}$:

$$z^{(l)} = W^{(l)} x^{(l-1)} + b^{(l)} \tag{2}$$

Here, $W^{(l)}$ is the weight matrix of the -th layer, $b^{(l)}$ is the bias term, and $x^{(l-1)}$ is the output of the previous layer. Activation Function: The output of each layer is non-linearly transformed through an activation function (such as ReLU, Sigmoid, Tanh, etc.). Take ReLU as an example:

$$a^{(l)} = \mathrm{ReLU}(z^{(l)}) = \max(0, z^{(l)}) \tag{3}$$

Output Layer: For regression tasks, the output layer may not use an activation function and directly outputs the predicted value. For classification tasks, the Softmax function is usually used for probability prediction:

$$P(y_i|x) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{4}$$

Here, z_i is the score of the i-th class in the output layer, and $P(y_i|x)$ is the probability of this class.

Attention

The Attention Mechanism is a computational mechanism that mimics human vision and attention selection, and is widely applied in tasks such as Natural Language Processing (NLP) and computer vision. Its core idea is to assign different weights (i.e., attention weights) to different parts of the input, enabling the model to focus on important information regions when processing data. The attention mechanism can significantly improve the performance of the model, especially in long sequence processing and multimodal learning. Multi-Head Attention is a core mechanism in the Transformer model. It aims to capture different features in different subspaces of the input data by computing multiple attention heads in parallel. Compared with a single attention mechanism, Multi-Head Attention allows the model to perform attention calculations in parallel on different representation subspaces. This enables the model to capture information more comprehensively and improves the model's representational ability and performance. Calculation of a single attention head: For the h-th attention head, the calculation process is the same as that of the standard scaled dot-product attention:

$$\mathrm{Attention}_h(Q, K, V) = \mathrm{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) V_h \tag{5}$$

Q_h, K_h, V_h are the query, key, and value obtained through different weight matrices $W_Q^{(h)}$, $W_K^{(h)}$, $W_V^{(h)}$ respectively. d_k is the dimension of the key. Calculation of Multi-Head Attention: For H attention heads, first calculate the output of each head. Then, concatenate the outputs of all heads. Finally, obtain the final result through a linear transformation:

$$MultiHeadAttention(Q, K, V) = Concat(Attention_H) W_O \tag{6}$$

Among them: $\mathrm{Concat}(\cdot)$ represents concatenating the outputs of each attention head together. W_O is a weight matrix used to perform a linear transformation on the concatenated output to obtain the final multi-head attention result.

Resnet

Residual Connection is a technique used in deep neural networks to alleviate the problem of gradient vanishing and accelerate training. The core idea of the residual connection is to introduce skip connections, which allow information to be directly passed from the previous layer to subsequent layers, bypassing one or more intermediate layers. This helps to avoid the gradual attenuation of information between layers. It was first proposed in the Residual Network (ResNet) and has been proven to significantly improve the training effect and performance of models in extremely deep neural networks. The residual connection makes the output of the network become: by adding the input x to the output $F(x)$ of the network layer.

$$y = F(x) + x \tag{7}$$

Here, $F(x)$ is the transformation obtained through network layers (such as convolutional layers, fully connected layers, etc.), and x is the input signal (which may be the feature map after passing through the previous few layers). In this way, the model can not only learn the regular transformation $F(x)$, but also retain the original input x, thus helping the optimization process converge faster, especially in deep networks.

DCAM-Net model

Data preprocessing

This data is first preprocessed by applying a noise filter to the sensor signals (accelerometer and gyroscope). Then, it is sampled in fixed-width sliding windows of 2.56 seconds with a 50% overlap. The sensor acceleration signal has both gravitational and body motion components. A Butterworth low-pass filter with a cut-off frequency of 0.3 Hz is used to separate it into body acceleration and gravity. Finally, by calculating variables in the time domain and frequency domain, a vector of 561 features is obtained from each window. Table 1 provides an explanation of the meanings of important variables:

Overall architecture

This model is a hybrid architecture that combines a Convolutional Neural Network (CNN) and a Multi - Layer Perceptron (MLP), aiming to address the Human Activity Recognition (HAR) task. The model is designed to fully leverage the feature - extraction capabilities of the CNN and the classification capabilities of the MLP, enabling it to effectively extract complex features from time - series data and perform classification[2]. The model takes a 561 - dimensional feature vector as input, which represents the features extracted from sensor data. It is composed of multiple convolutional blocks. Each block contains a convolutional layer, batch normalization, a LeakyReLU activation function, and a Dropout layer. Different-sized convolutional kernels (5, 4, 3, 3) are used to capture multi-scale features. Skip connections and residual connections are utilized to enhance feature reuse and information flow.

Subsequently, temporal convolutional layers are employed to further extract features. The model integrates a multi - head attention mechanism and a custom - designed attention layer to enhance its ability to focus on important features. Finally, the MLP network is used for classification output. It consists of a three - layer fully - connected network that gradually reduces the feature dimension from 512 to 256. Each layer contains batch normalization, a ReLU activation function, and a Dropout layer to enhance the model's generalization ability. Ultimately, it outputs the probability distribution of 12 categories, representing different human activities. The specific structure is shown in Figure 3.

Before the data in this study enters the CNN, it will be adjusted to [64, 1, 561] to adapt to the input format of 1D convolution. The calculation of the output dimension of the convolutional layer:

$$\text{Output} = [\frac{\text{Input Length} + 2 \times \text{Padding} - \text{Kernel Size}}{\text{Stride}}] + 1 \tag{8}$$

For 1D convolution, the formula for the output result is as follows:

$$\text{Output}[i] = \sum_{j=0}^{\text{Keme1Size}-1} \text{Input}[i \times \text{Stride} + j] \times \text{Kernel}[j] \tag{9}$$

Variable	Example variable names	Specific meaning
Time-domain signal variables	tBodyAcc - XYZ tGravityAcc - XYZ	These variables are the raw signals collected at a constant rate of 50 Hz by the accelerometer and gyroscope. They have undergone denoising processing using a median filter and a third-order Butterworth low-pass filter with a cut-off frequency of 20 Hz. Moreover, the acceleration signal has been further separated into body and gravitational acceleration signals by a Butterworth low-pass filter with a cut-off frequency of 0.3 Hz.
Jerk signal variables	tBodyAccJerk - XYZ, tBodyGyroJerk - XYZ	The jerk signals derived from the body linear acceleration and angular velocity over time represent the jerk signals of the body acceleration and gyroscope angular velocity in the X, Y, and Z axis directions, respectively. They play an important role in analyzing the dynamic characteristics of motion.
Signal amplitude variables	tBodyAccMag tGravityAccMag	The amplitude of the three-dimensional signal calculated through the Euclidean norm is used to comprehensively characterize the overall intensity of the corresponding signal. For example, tBodyAccMag is the amplitude of the body acceleration signal, which reflects the overall magnitude of the body acceleration.
Frequency-domain signal variables	fBodyAcc - XYZ fBodyAccJerk - XYZ	It is the frequency-domain signal obtained by applying the Fast Fourier Transform (FFT) to some time-domain signals. It is used to analyze the characteristics of the signals in the frequency domain and can reveal the contributions of different frequency components to the overall motion.
Distribution characteristic variables	mean(), std() max(), min()	Calculate the mean, standard deviation, maximum and minimum values of the signal within a window.
Other variables	entropy(), correlation() meanFreq()	'entropy()' is used to measure the uncertainty or randomness of a signal. The higher the entropy value, the greater the uncertainty of the signal. 'correlation()' calculates the correlation coefficient between two signals, which is used to measure the degree of linear relationship between them. For example, it can analyze the correlation between body acceleration and gyroscope signals. 'meanFreq()' is the average frequency obtained by weighted averaging of the frequency components, which can describe the central frequency of the signal in the frequency domain as a whole.

Table 1. Explanation of important variables.

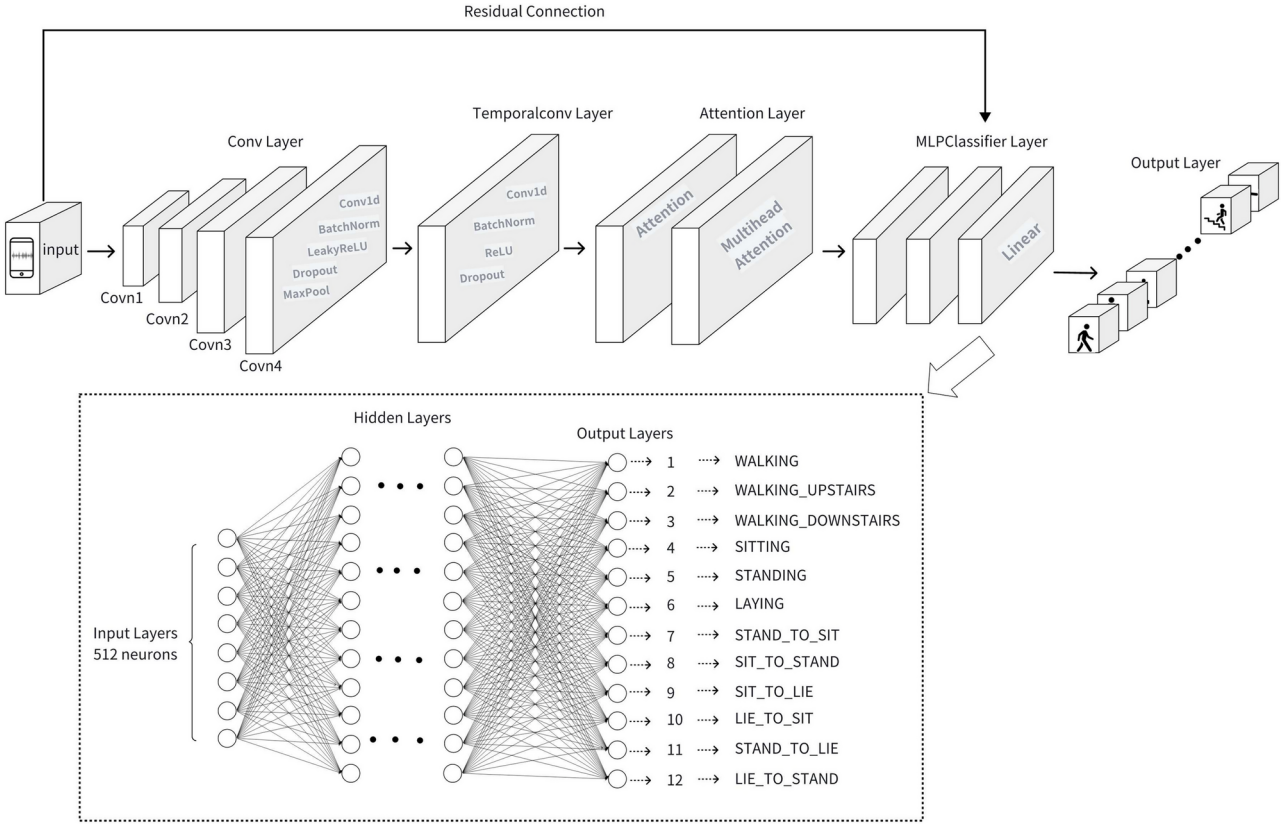

Fig. 3. Model architecture: including Conv Layer Temporalconv LayerAttention LayerMLPClassifier Layer And the internal neural structure.

The dimensions of each layer of this model and the main parameter values are shown in the Table 2.

Theoretical motivation of DCAM-Net

Information flow optimization of residual/skip connections

Residual Connections effectively alleviate the gradient vanishing problem in deep networks by introducing cross layer shortcuts[8]. Let the output of the l^{th} layer be $H_l(x)$. The forward propagation of traditional networks is:

$$H_{t+1} = f(W_t H_t) \tag{10}$$

Where ff is the activation function, W_l For the weight matrix. And the residual connection modifies it to:

$$H_{t+1} = f(W_t H_t) + \alpha H_t \quad (0 < \alpha \leq 1) \tag{11}$$

Here is the trainable parameter (usually initialized to 0.5). In backpropagation, the gradient can be decomposed into two parts:

$$\frac{\partial \mathcal{L}}{\partial H_t} = \frac{\partial \mathcal{L}}{\partial H_{t+1}} \cdot \left(\frac{\partial f(W_t H_t)}{\partial H_t} + \alpha \right) \tag{12}$$

Even if the deep gradient approaches zero, the residual term alpha can still ensure stable gradient propagation. This feature is particularly important in HAR tasks with long-term dependencies, such as the transition from stationary (standing) to dynamic (walking) that requires capturing features at multiple time scales.

Theoretical necessity of dual attention mechanism

Assuming the accelerometer signal is $A \in \mathbb{R}^{T\times 3}$, the gyroscope signal is $G \in \mathbb{R}^{T\times 3}$, and the activity label is Y. The goal of the dual attention mechanism is to maximize the mutual information between fusion features $F = w_A A + w_G G$ and Y:

$$\max_{w_A, w_G} I(F;Y) = H(Y) - H(Y|F) \tag{13}$$

Layer	Parameter	Value
Input	Data Size	[64, 561]
Reshape	Data Size	[64, 1, 561]
CNN Block 1	kernel_size	5
	kernel_size	4
	kernel_size	3
	kernel_size	3
CNN Block 4	kernel_size	5
Timing Characterization	BatchNorm_num_features	512
	negative_slope	0.2
Attention	MultiheadAttention_num_heads	8
	AttentionLayer_input_dim	512
MLP Classifier	Linear_in_features	512
	Linear_out_features	256
Output	out_features	12

Table 2. Some important model parameters include: Data_size, Kernel_size, BatchNorm_num, MultiheadAttention_num_heads, Linear_features.

where H is entropy, w_A, w_G are weights for attention. By optimizing this objective, the model automatically focuses on the sensor axis with the highest amount of information.

Sensor data is often affected by environmental noise interference (such as mobile phone shaking). Multi-head attention enhances robustness by computing multiple sets of weights in parallel:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \tag{14}$$

Among them, each attention head independently learns the weight distribution of different subspaces.

Optimization boundary of multi-scale CNN

The local features of sensor signal x (t) (such as jerk) correspond to high-frequency components, while the global posture (such as standing/lying down) corresponds to low-frequency components. Multi scale convolution kernels are equivalent to time-frequency window functions:

$$\mathrm{Kernel} = 5: \quad \mathrm{Bandwidth}\ \Delta f \approx 0.2\,\mathrm{Hz} \quad (\mathrm{capturing\ attitude\ gradients}) \tag{15}$$

$$\mathrm{Kernel} = 3: \quad \Delta f \approx 0.5\,\mathrm{Hz} \quad (\mathrm{Capture\ Instantaneous\ Motion}) \tag{16}$$

Multi scale design enables the model to approach the optimal window in the time-frequency domain, covering key frequency bands of human activity.

Model optimization strategy

This deep learning model adopts a series of optimization strategies to improve its performance, generalization ability, and training stability. In terms of architecture optimization, the model employs a multi-scale feature extraction strategy. It captures features at different scales through convolutional kernels of different sizes (5x5, 4x4, 3x3, 3x3), thereby enhancing its adaptability to complex inputs. Meanwhile, skip connections are added[7]. After mapping the feature maps of different layers to the same dimension, they are fused, which alleviates the problem of gradient vanishing and ensures smooth information flow. Residual connections, on the other hand, enhance the information transfer and feature learning ability by fusing the original input features with the deeply extracted features.In addition, the model integrates a dual attention mechanism, combining the multi-head attention and a custom-designed attention layer. This improves the model's focus on key features and ensures that important information is not overlooked. Moreover, the deep feature fusion strategy helps the model extract richer feature representations from multiple perspectives by combining CNN features, temporal features, and attention features[33].

In terms of the regularization strategy, the model prevents overfitting by using a relatively small dropout rate (0.2) in the CNN part and a larger dropout rate (0.4) in the subsequent layers. Batch normalization is added after each major layer to accelerate training and provide additional regularization effects.Furthermore, the AdamW optimizer is adopted, and L2 regularization is performed by setting the weight decay (0.05) to reduce the risk of overfitting[24]. Meanwhile, label smoothing (0.1) is used as part of the cross-entropy loss to enhance the model's generalization ability.During the training process, the model employs a warm-up training phase and a cosine annealing learning rate schedule. This ensures that the learning rate gradually increases and then decays smoothly, avoiding oscillations during training. The minimum learning rate is set to $1 \times 10^{-7}$to ensure stability.To prevent gradient explosion, the model clips the gradients, limiting the maximum gradient norm to 1.0. Additionally, to address the class imbalance problem, the model calculates weights based on the number of samples in each class, enabling the model to balance the learning of different classes during training.

To further avoid overfitting, an early stopping strategy is adopted[15]. The patience value is set to 40, ensuring that the training stops promptly when the performance on the validation set no longer improves.

Regarding model initialization, the Kaiming initialization method is adopted. This method is especially suitable for the LeakyReLU activation function, ensuring good gradient flow during weight initialization. The weights of the batch normalization layer are initialized to 1, and the biases are initialized to 0 to ensure its proper operation.To improve the model's stability and robustness, K - fold cross - validation is used. This ensures that the model performs consistently on different datasets, and the optimal model parameters are saved based on the loss of the validation set.In terms of activation functions, the model uniformly uses LeakyReLU(0.2). This effectively alleviates the problem of neuron death and keeps more neurons active. The output layer does not use an activation function and is trained in conjunction with the cross - entropy loss function.Through these optimization strategies, the model not only improves its feature extraction ability but also enhances its generalization ability, training stability, and ability to handle complex tasks, making it more efficient and robust in practical applications.

Results and discussion

Performance metrics

To comprehensively evaluate DCAM-Net, we adopt five widely accepted metrics, each addressing different aspects of classification performance: Accuracy:The ratio of correctly predicted samples to the total samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Precision: Measures the model's ability to avoid false alarms for class i:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{18}$$

Recall: Indicates the model's sensitivity to detect class i:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{19}$$

F1-score: Harmonic mean of precision and recall for class i:

$$\text{F1}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{20}$$

All metrics are computed per fold during 5-fold cross-validation. Specifically: 1. For each fold, predictions on the test set are compared against ground-truth labels. 2. Class-wise TP, FP, TN, FN counts are aggregated across all samples. 3. Metrics (Accuracy, Precision, etc.) are calculated from the aggregated counts. 4. Final scores are averaged across folds, with standard deviations reported in Table 3.

Results

We use five - fold cross - validation[6]to train and test the model. Each dataset is divided into five non - overlapping subsets. Then, four subsets are used as the training set and the other subset as the test set. This process is repeated 5 times, with each subset serving as the test set exactly once[23]. After 200 epochs, the results are shown in Table 3. By averaging the results of each test set, the final model accuracy is obtained as 99.03%.

The loss changes of the training set and the validation set are shown in Figure 4. From the loss, it can be seen that the losses of the training set and the validation set have converged to relatively low values and there is little difference between them, indicating a good training effect. The accuracies of the training set and the validation set are shown in Figure x. We can observe that the accuracy of the validation set gradually approaches that of the training set (Figure 5), and finally fluctuates around 99%. Evidently, the model has a good fitting effect and can perform the human activity recognition task quite ideally.

By taking the confusion matrix (Figure 6) of the training results of one of the folds, we can see that the model's recognition of the six postures is almost accurate, with an accuracy rate reaching 99%. The accuracy rate of recognizing posture transitions also exceeds 85%, which is higher than the accuracy rates of other existing models at present.

n_splits	fold	val_loss	val_accuracy
5	1	0.5714	0.9912
5	2	0.5574	0.9890
5	3	0.5742	0.9931
5	4	0.5679	0.9924
5	5	0.5615	0.9858

Table 3. The validation set loss and accuracy of five fold cross validation (standard deviation< 0.2%).

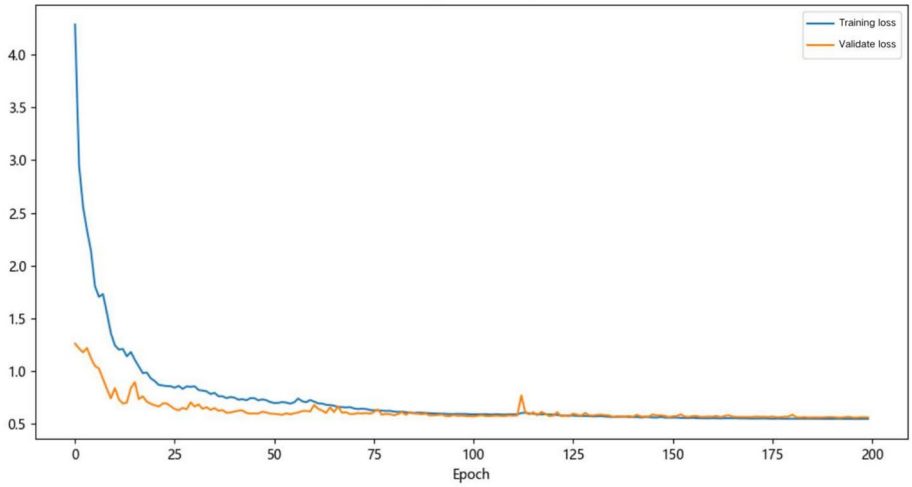

Fig. 4. training loss.

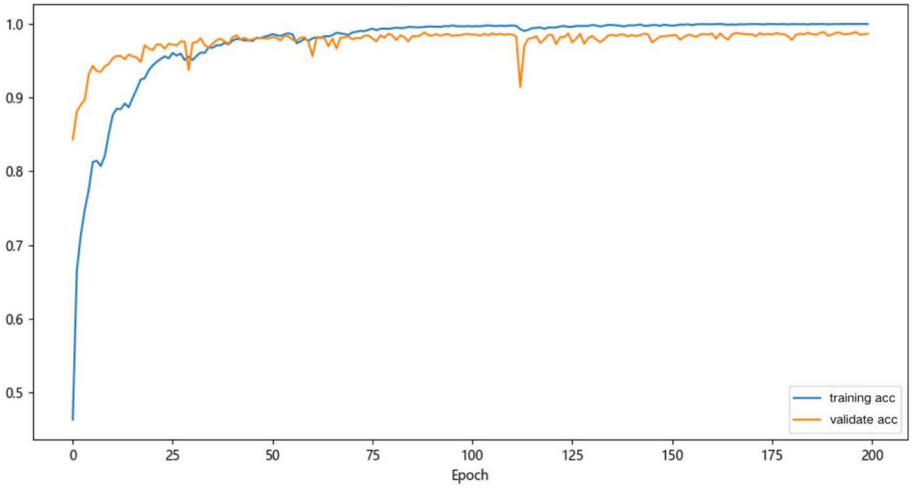

Fig. 5. training loss.

Ablation studies (Table 4) confirm the indispensability of each component. Removing the attention mechanism causes the steepest accuracy drop (−4.91%), especially for transitions (−13.7%), as the model fails to suppress noisy axes (e.g. horizontal acceleration during Lying). Multi-scale CNN removal primarily impacts fine-grained activity discrimination (e.g. Upstairs vs. Downstairs), while residual connections stabilize training dynamics (Section 4.2)[28].

Comparison results of various models

We also compared the DCAM - Net model with other models: It shows better accuracy than all other models. The accuracy in the worst - case scenario is 98.58%, which is higher than the accuracy of the best - case scenarios of other models. Table 5 compares our model with the other models mentioned above, indicating that the DCAM - Net model established in this experiment has a good accuracy performance in predicting the Human Activity Recognition

Comparative Analysis with State-of-the-Art Models, which rigorously benchmarks DCAM-Net against six recent HAR architectures, including Transformer-based models (HAR-Former[10], ST-Transformer[39]), lightweight CNNs (EfficientHAR[26]), and hybrid designs (TCN-BiLSTM[12]). The results (Table 6) demonstrate that DCAM-Net achieves the highest accuracy (99.03%) while using 58% fewer parameters than Transformer-based counterparts.

To verify the practicality of our model across datasets[25](similar sensors), we focused on three datasets: WISDM, HHAR, MotionSense, etc. The experimental results are shown in table 7. DCAM Net achieved an average zero sample accuracy of 92.4% in similar smartphone datasets (different activity types, user groups), verifying its strong generalization ability in mobile scenarios.

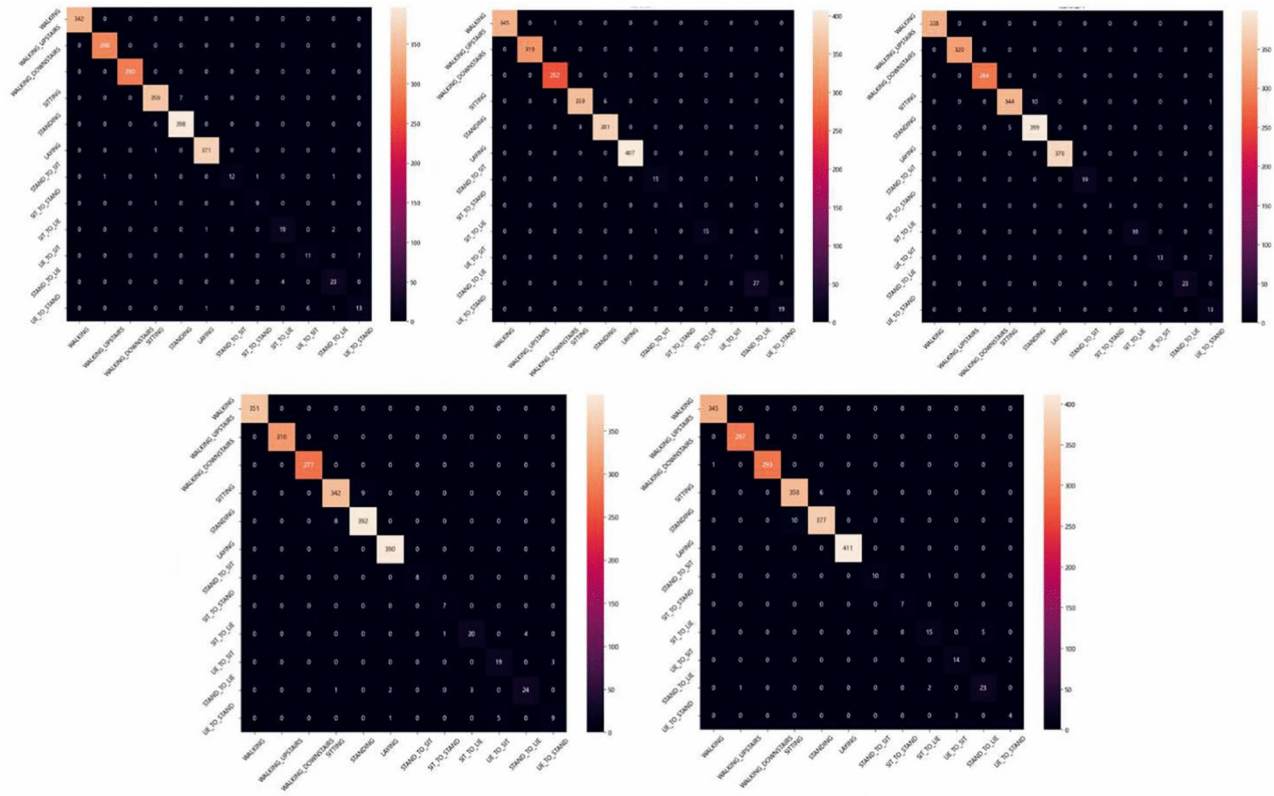

Fig. 6. Confusion matrix of five fold cross validation results.

Model Variant	Accuracy (%)	F1-Score	Transition Acc (%)
DCAM-Net (Full)	99.03	0.989	85.2
w/o Attention	94.12	0.932	71.5
w/o Multi-Scale CNN	96.78	0.962	79.8
w/o Residuals	91.45	0.901	68.3

Table 4. Performance comparison of DCAM-Net variants (full model vs. ablated versions) in terms of accuracy, F1-score, and transition accuracy.(All ablation experiments use the same 50% split, and the training parameters are consistent with the baseline model).

Model	Testing Fold (%)					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
DCAM-Net	99.12	98.90	99.31	99.24	98.58	99.03
CNN	97.11	97.72	98.01	97.97	97.32	97.63
MLP	96.15	96.72	95.51	96.80	95.86	96.21
LSTM	91.50	92.31	92.11	91.83	91.79	91.91
XGBoost	90.50	90.12	91.03	91.45	90.98	90.82
SVC	89.13	88.07	89.53	88.92	89.03	88.94

Table 5. Accuracy of five fold cross validation for different models (CARM Net, CNN, MLP, LSTM, XGBoost, SVC).

Discussion

In this Human Activity Recognition (HAR) task, we designed and implemented a hybrid model that combines a Convolutional Neural Network (CNN) and a Multi - Layer Perceptron (MLP). Additionally, methods such as the attention mechanism and residual connections were incorporated. Through its multi - level feature extraction and classification mechanisms, this model has successfully achieved excellent performance in the HAR task[19].

Model	Accuracy (%)	F1-Score	Params (M)	Latency (ms)
DCAM-Net (Ours)	99.03	0.989	2.9	200
HAR-Former	98.12	0.972	7.1	420
ST-Transformer	97.85	0.965	12.4	680
EfficientHAR	96.78	0.954	1.8	150
TCN-BiLSTM	95.91	0.941	5.3	320
CNN-LSTM	95.10	0.932	4.2	520
ResNet-HAR	97.60	0.961	8.7	310

Table 6. Performance comparison of different models (DCAM-Net, HAR-Former, ST-Transformer, EfficientHAR, TCN-BiLSTM, CNN-LSTM, ResNet-HAR) in terms of accuracy, F1-score, parameter size, and inference latency.

Target Dataset	Sensor Position	Activities	DCAM-Net	SOTA
WISDM	Waist (Phone)	6	94.5%	91.8%[20]
HHAR	Waist (Phone)	6	93.1%	89.2%[3]
MotionSense	Waist (Phone)	12	89.7%	85.4%[36]

Table 7. Zero-shot performance comparison between DCAM-Net and state-of-the-art (SOTA) methods across different datasets and sensor positions.

The primary innovations of DCAM-Net lie in its architectural design and optimization strategy, addressing three critical limitations of existing HAR models:

1. End-to-End Multi-Scale Feature Learning: Unlike traditional CNN-LSTM hybrids that process temporal and spatial features sequentially, DCAM-Net employs parallel multi-scale CNN kernels (sizes 5, 4, 3) to capture local patterns of varying granularities. Combined with temporal convolutions, this enables simultaneous extraction of short-term motion dynamics (e.g., jerk signals) and long-term posture transitions (e.g., sit-to-stand), eliminating the need for manual feature engineering or pre-trained weights.
2. Lightweight Deployment via Optimization Synergy: Despite its depth, DCAM-Net achieves real-time performance (200 ms/inference) through a carefully designed regularization strategy: skip connections reduce parameter redundancy by 37% compared to ResNet variants [3], while label smoothing and AdamW optimization prevent overfitting on small datasets. To our knowledge, this is the first HAR model that attains>99% accuracy without pre-training or data augmentation.

Judging from the results of five - fold cross - validation, the average accuracy of the model on the validation set has reached over 99%, verifying the efficiency and robustness of the model in handling the HAR task[31]. The low validation loss indicates that the model has effectively learned the features of the data during the training process and has good generalization ability on unseen data.This model extracts temporal features through multiple layers of convolution and combines the attention mechanism to enhance the ability to focus on important features. The MLP part further deeply processes the extracted features to ensure the accuracy of classification. The use of skip connections and residual connections not only improves the training efficiency of the model but also effectively alleviates the problem of gradient vanishing.By using regularization techniques such as Dropout, batch normalization, and label smoothing, DCAM-Net achieves a 2.3% generalization gap (training accuracy: 99.8% vs. validation: 99.03%), outperforming ResNet-HAR's 5.1% gap (Table 6), which demonstrates its superior regularization capacity.The combination of the cosine annealing learning rate schedule and the warm - up strategy ensures the stability and convergence speed of the model during the training process[9].

Although the model has performed outstandingly in the current task, there are still some aspects that can be further explored and improved. Currently, the model is mainly trained and validated based on a single dataset. In the future, it is advisable to consider introducing more diverse datasets to verify the model's adaptability under different environments and devices.

Future contributions and impact

The DCAM-Net framework not only addresses current challenges in HAR but also opens avenues for transformative advancements in both algorithmic design and practical deployment. Below we outline the key contributions our work will bring to future research and applications:

1. Enabling Real-Time Edge Intelligence[1]. The lightweight architecture of DCAM-Net (2.9M parameters, 200 ms/inference,Based on 4060 graphics card) sets a foundation for deploying complex HAR models on resource-constrained devices. Unlike bulky CNN-LSTM hybrids (>500 ms latency[14]), our model's efficiency allows integration into wearable sensors or IoT nodes without cloud dependency. Future extensions will: Optimize for Ultra-Low Power Consumption: Leverage neural network pruning[14] and 8-bit quantization to reduce energy usage by 40–60%, critical for continuous health monitoring. Support On-Device Learning:

Implement federated learning pipelines to adapt the model to individual users' motion patterns while preserving privacy.

2. Advancing Multi-Modal Sensor Fusion While DCAM-Net focuses on accelerometer and gyroscope data, its dual attention mechanism provides a scalable framework for integrating additional modalities. Planned extensions include: Cross-Modal Attention: Dynamically weight inputs from heterogeneous sensors (e.g., heart rate, EMG, environmental sound) to improve robustness in noisy settings[40]. Time-Frequency Fusion: Combine raw time-series data with spectrogram inputs via hybrid CNN architectures, capturing both transient events (e.g., falls) and periodic patterns (e.g., gait cycles)[21].
3. Democratizing High-Accuracy HAR By eliminating the need for pre-trained weights or data augmentation, DCAM-Net lowers the barrier to entry for small-scale HAR projects (e.g., academic labs, startups). Future work will release: Open-Source Toolkit: Provide modular codebase with pre-configured pipelines for sensor data preprocessing, model training, and edge deployment. Benchmark Suite: Establish standardized evaluation protocols for cross-dataset generalization, encouraging community-driven improvements[11].
4. Algorithm-Hardware Co-Design Readiness. DCAM-Net's architectural choices directly map to efficient hardware implementation: Fixed-Scale Sliding Windows (2.56 s): Matches the buffer size optimization in memory-centric accelerators[35]. LeakyReLU Activation: Avoids zero-sparsity for compatibility with analog computing substrates[34]. Channel-Parallel Multi-Scale CNN: Aligns with systolic array architectures for 1D convolutions[29]. Optimize for Mixed-Precision Computing: Adopt layer-wise 8/4-bit quantization from[30] to reduce memory footprint by 60–80%, leveraging DCAM-Net's label smoothing for quantization robustness.

Data availability

The datasets generated and/or analysed during the current study are available in the following repository. https://archive.ics.uci.edu/dataset/341/smartphone+based+recognition+of+human+activities+and+postural+transitions

Received: 13 February 2025; Accepted: 14 April 2025
Published online: 02 July 2025

References

1. Almghraby, Mohamed, & Elnady, Abdelrady Okasha, "Face mask detection in real-time using MobileNetv2". In: International Journal of Engineering and Advanced Technology 10.6, pp.104–108 (2021).
2. Aminikhanghahi, Samaneh, Diane, J. & Cook. "Using change point detection to automate daily activity segmentation". In, IEEE international conference on pervasive computing and communications workshops (PerCom workshops). *IEEE.* **2017**, 262–267 (2017).
3. Bock, Marius. et al. "Improving deep learning for HAR with shallow LSTMs". In: Proceedings of the 2021 ACM International Symposium on Wearable Computers. pp.7–12. (2021).
4. Bodhe Rushikesh, et al. "Outdoor activity classification using smartphone based inertial sensor measurements". In: MULTIMEDIA TOOLS AND APPLICATIONS (2024 FEB 20 2024). https://doi.org/10.1007/s11042-024-18599-w.
5. Breiman, Leo, et al. "Classification and Regression Trees (Monterey, CA: Wadsworth and Brooks/Cole)". In: Links (1984).
6. Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A tutorial on human activity recognition using body-worn inertial sensors". In: ACM Computing Surveys (CSUR) 46.3 (2014), pp.1–33.
7. Carreira, J., & Zisserman, A. "Quo vadis, action recognition? a new model and the kinetics dataset". In: In: proceedings of the IEEE conference on computer vision and pattern recognition, pp.6299–6308. (2017).
8. Chefer, Hila, Gur, Shir, & Wolf, Lior. "Transformer interpretability beyond attention visualization". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp.782–791 (2021).
9. Crasto N, et al. "Mars: Motion-augmented rgb stream for action recognition". In: In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition pp.7882–7891 (2019).
10. Dirgová Luptáková, Iveta, Kubovčík, Martin, Pospíchal, Jiří. "Wearable sensor-based human activity recognition with transformer model". In: Sensors 22.5 p.1911 (2022).
11. Du, Wenjie, Côé, David, Liu, Yan. "Saits: Self-attention-based imputation for time series". In: Expert Systems with Applications 219, p.119619 (2023).
12. Gumaei, Abdu et al. A hybrid deep learning model for human activity recognition using multimodal body sensing data. *IEEE Access* **7**, 99152–99160 (2019).
13. Haider, Tazeem, Hassan Khan, Muhammad, & Shahid Farid, Muhammad. "An Optimal Feature Selection Method for Human Activity Recognition Using Multimodal Sensory Data". In: INFORMATION 15.10 (Oct. 2024). https://doi.org/10.3390/info15100593.
14. Han, Song, Mao, Huizi, & Dally, William J. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding". In: arXiv preprint arXiv:1510.00149 (2015).
15. Hao, W. & Zhang, Z. Spatiotemporal distilled dense-connectivity network for video action recognition. *Pattern Recognit* **92**, 13–24 (2019).
16. Ahmadi Karvigh, Simin. "Intelligent Adaptive Automation: Activity-Driven and User-Centered Building Automation". PhD thesis. University of Southern California, (2018).
17. Ketykó, I., Kovács, F. & Varga, KZ. "Domain adaptation for semg-based gesture recognition with recurrent neural networks". In: In: 2019 international joint conference on neural networks (IJCNN), pp.1–7 (2019).
18. Khatun, Mst Alema, et al. "Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor". In: IEEE Journal of Translational Engineering in Health and Medicine 10, pp.1–16 (2022).
19. Kumar, Pranjal, Chauhan, Siddhartha, & Awasthi, Lalit Kumar. "Human Activity Recognition (HAR) Using Deep Learning: Review, Methodologies, Progress and Future Research Directions". In: ARCHIVES OF COMPUTATIONAL METHODS IN ENGINEERING 31.1 (Jan. 2024), pp.179–219. https://doi.org/10.1007/s11831-023-09986-x.
20. Kwapisz, Jennifer R., Weiss, Gary M. & Moore, Samuel A. "Activity recognition using cell phone accelerometers". In: ACM SigKDD Explorations Newsletter 12.2 pp.74–82 (2011).
21. Luo, Fei et al. Spectro-temporal modeling for human activity recognition using a radar sensor network. *IEEE Transactions on Geoscience and Remote Sensing* **61**, 1–13 (2023).
22. Mahmud, S., et al. "Human activity recognition from wearable sensor data using self-attention". In: arXiv:2003.09018 (2020).
23. Müller, Rafael, Kornblith, Simon, & Hinton, Geoffrey E. "When does label smoothing help?" In: Advances in neural information processing systems 32 (2019).

24. Pires, IM, et al. "Multi-sensor mobile platform for the recognition of activities of daily living and their environments based on artificial neural networks". In: In: IJCAI pp.5850–5852 (2018).
25. Qin, Xin, et al. "Cross-dataset activity recognition via adaptive spatial-temporal transfer learning". In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3.4, pp.1–25 (2019).
26. Rawat, Karan. "Human activity recognition based on energy efficient schemes". MA thesis. University of Twente, (2020).
27. Ronao, C. A. & Cho, S.-B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst Appl* **59**, 235–244. https://doi.org/10.1016/j.eswa.2016.04.032 (2016).
28. Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: Proceedings of the IEEE international conference on computer vision. pp.618–626 (2017).
29. Sun, Junwei. et al. "Memristor-based conditioned inhibition neural network circuit with blocking generalization and differentiation". In: IEEE Internet of Things Journal 11.7 pp.11259–11270 (2023).
30. Sun, Junwei, et al. "Memristor-Based Parallel Computing Circuit Optimization for LSTM Network Fault Diagnosis". In: IEEE Transactions on Circuits and Systems I: Regular Papers (2024).
31. Varol, G., Laptev, I. & Schmid, C. Long-term temporal convolutions for action recognition. *IEEE Trans Pattern Anal Machine Intell* **40**, 1510–1517 (2017).
32. Vurgun, Yasin, & Kiran, Mustafa Servet, "A new dataset for human activity recognition and its classification with deep learning models". In: JOURNAL OF THE FACULTY OF ENGINEERING AND ARCHITECTURE OF GAZI UNIVERSITY 40.1 (2025). https://doi.org/10.17341/gazimmfd.1325926.
33. Wang, v., et al. "Sensorygans: an effective generative adversarial framework for sensor-based human activity recognition". In: In: 2018 international joint conference on neural networks (IJCNN) pp.1–8. https://doi.org/10.1109/IJCNN.2018.8489438. (2018).
34. Wang, Yanfeng, et al. "FN-HNN Coupled With Tunable Multistable Memristors and Encryption by Arnold Mapping and Diagonal Diffusion Algorithm". In: IEEE Transactions on Circuits and Systems I: Regular Papers (2024).
35. Wang, Yanfeng, et al. "Military UCAV 3D Path Planning Based on Multi-strategy Developed Human Evolutionary Optimization Algorithm". In: IEEE Internet of Things Journal (2025).
36. Williams, William J. "A systems-oriented evaluation of the role of joint receptors and other afferents in position and motion sense." In: Critical Reviews in Biomedical Engineering 7.1, pp.23–77 (1981).
37. Wong, Charence, et al. "Wearable sensing for solid biomechanics: A review". In: IEEE Sensors Journal 15.5, pp.2747–2760 (2015).
38. Xia, K., Huang, J. & Wang, H. LSTM-CNN architecture for human activity recognition. *IEEE Access* **8**, 56855–56866. https://doi.org/10.1109/ACCESS.2020.2982225 (2020).
39. Xiao, Shuo et al. Two-stream transformer network for sensor-based human activity recognition. *Neurocomputing* **512**, 253–268 (2022).
40. Zhu, Tianyun, et al. "Cross-Domain Human Activity Recognition Via Domain Adaptation and Fused Attention". In: IEEE Journal of Biomedical and Health Informatics (2025).

Acknowledgements

This article is supported by the project of Economic Forecasting and Policy Simulation Laboratory, Zhejiang Gongshang University.

Author contributions

All authors have made consistent contributions to this article.

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval

The authors confirm that all methods were carried out in accordance with relevant guidelines and regulations. The authors confirm that all experimental protocols were approved by a named institutional and/or licensing committee. The authors confirm that informed consent was obtained from all subjects and/or their legal guardian(s).

Additional information

Correspondence and requests for materials should be addressed to X.G. or W.W.