computers

MDPI

Article

Human Activity Recognition Using Graph Structures and Deep Neural Networks

Abed Al Raoof K. Bsoul

Collage of Computer Science and Information Technology, Yarmouk University, Irbid 21163, Jordan; raoofbsoul@yu.edu.jo

Abstract: Human activity recognition (HAR) systems are essential in healthcare, surveillance, and sports analytics, enabling automated movement analysis. This research presents a novel HAR system combining graph structures with deep neural networks to capture both spatial and temporal patterns in activities. While CNN-based models excel at spatial feature extraction, they struggle with temporal dynamics, limiting their ability to classify complex actions. To address this, we applied the Firefly Optimization Algorithm to fine-tune the hyperparameters of both the graph-based model and a CNN baseline for comparison. The optimized graph-based system, evaluated on the UCF101 and Kinetics-400 datasets, achieved 88.9% accuracy with balanced precision, recall, and F1-scores, outperforming the baseline. It demonstrated robustness across diverse activities, including sports, household routines, and musical performances. This study highlights the potential of graph-based HAR systems for real-world applications, with future work focused on multi-modal data integration and improved handling of occlusions to enhance adaptability and performance.

Keywords: human activity recognition (HAR); deep learning; firefly optimization algorithm; graph-based models; spatial–temporal analysis

1. Introduction

Human activity recognition (HAR) has emerged as a critical area of research within the domain of computer vision and machine learning, owing to its vast potential applications in healthcare, smart surveillance, and human–computer interactions. HAR aims to identify and classify human activities based on sensor data or video streams, thereby enabling machines to interpret and respond to human actions in real time. HAR systems can provide valuable insights for various intelligent applications. This technology has been widely applied in fields such as home behavior analysis, video surveillance, gait analysis, and gesture recognition.

HAR systems rely on data from active sensors or passive sensors. Active sensors emit energy in the form of electromagnetic waves or sound to detect objects or measure environmental conditions. The sensors then analyze the reflected signals to gather data. On the other hand, passive sensors do not emit their own signals but instead detect natural energy or signals present in the environment. These sensors measure ambient conditions or capture data from external sources of energy, such as light or sound. Examples of active sensors are accelerometer, gyroscope, radar, lidar, and Kinect sensors. Optical cameras, infrared cameras, microphones, and environmental sensors are considered examples of passive sensors.

Due to the rapid development of sensor technology and ubiquitous computing, sensor-based HAR has gained popularity, offering the advantage of privacy protection. Active sensor-based methods have demonstrated considerable success but often require users to

check for updates

Academic Editor: Paolo Bellavista

Received: 4 December 2024
Revised: 26 December 2024
Accepted: 29 December 2024
Published: 30 December 2024

Citation: Bsoul, A.A.R.K. Human Activity Recognition Using Graph Structures and Deep Neural Networks. *Computers* **2025**, *14*, 9. https://doi.org/10.3390/computers14010009

wear multiple sensors, which can be inconvenient and intrusive. Moreover, active sensor-based approaches are susceptible to issues like sensor misalignment and varying sensitivity, which can impact the accuracy of activity recognition. The discomfort associated with wearing sensors and the inherent limitations of sensor power necessitate the exploration of alternative methods. Passive sensor-based methods are considered the most comfortable HAR method since there is no direct interaction with humans. However, passive sensors may provide less precise or detailed data compared to active sensors, which can be a limitation in applications requiring high accuracy or detailed analysis.

HAR technologies can be broadly categorized into two types according to their location: approaches based on fixed sensors and mobile-based approaches.

Fixed sensor-based methods obtain information from sensors mounted at a specified position, including acoustic sensors, radars, and other ambient-based sensors. Among these, camera-based methods are the most popular, employing techniques such as background subtraction, optical flow, and energy-based segmentation to extract features. For example, image processing methods based on Kinect sensors can acquire depth image features of moving targets. Although these activity monitoring methods can provide high recognition accuracy, they are not suitable for many indoor environments, especially where privacy is a concern.

The other category of activity recognition methods involves using mobile sensors. In these methods, information from various behaviors is collected using dedicated body-worn motion sensors, such as accelerometers, gyroscopes, and magnetometers. These sensors detect changes in acceleration and angular velocity corresponding to human motion, allowing for the inference of activities. The miniaturization and flexibility of sensors enable individuals to wear or carry mobile devices embedded with various sensing units, distinguishing this approach from fixed sensor-based methods. Mobile sensors are characterized by low cost, low power consumption, high capacity, miniaturization, and reduced dependence on surroundings. Consequently, activity recognition based on mobile sensors has garnered widespread attention due to its portability and high acceptance in daily life. However, activity recognition using mobile sensors faces challenges such as high power consumption, user comfort issues, and privacy concerns. Environmental interference, inconsistent sensor placement, and lack of contextual awareness further affect accuracy. Large data volumes and sensor calibration issues also pose difficulties. Ongoing research is needed to improve reliability and user acceptance through advanced algorithms and integration with additional data sources. These limitations underscore the necessity for ongoing research to enhance the accuracy, reliability, and user acceptance of mobile sensor-based activity recognition systems by integrating additional data sources, advancing sensor technology, and developing sophisticated algorithms.

Recognizing the limitations of traditional sensor-based methods, researchers have turned to deep learning techniques to improve the performance and robustness of HAR systems. Deep learning techniques have revolutionized HAR by leveraging raw data from video streams captured by surveillance cameras. These methods utilize convolutional neural networks (CNNs) or Recurrent Neural Networks (RNNs) to automatically extract and learn features from video data, eliminating the need for manual feature engineering. This approach has proven particularly effective in recognizing complex activities and distinguishing between normal and anomalous behaviors, making it highly applicable in areas such as elderly care, autism monitoring, and public safety. The automatic feature extraction capability of deep learning models significantly reduces the dependency on domain experts and noisy data and enhances the scalability of HAR systems.

Despite the advancements brought by deep learning, challenges remain in effectively capturing the spatial and temporal relationships inherent in human activities. Recognizing

human activities from video data requires understanding not only the individual frames but also the transitions between frames, which encode crucial temporal information.

In this research, we propose a novel HAR system that integrates graph structures and deep neural networks to address these challenges. By representing human joint movements as a graph, where each node corresponds to a joint and edges represent the connections between them, the system can capture the intricate patterns of human motion more comprehensively. This graph-based representation allows for a more nuanced understanding of the spatial relationships and temporal dynamics involved in human activities, which is of paramount importance for any HAR system.

The primary objective of this research is to investigate the effectiveness of using graph structures derived from human joint 3D trajectories in enhancing HAR. The system will be implemented using the OpenPose algorithm to determine the optimal approach for recognizing actions. OpenPose is a state-of-the-art method for human pose estimation that provides precise joint location data, which can be used to construct graph representations of human activities. By focusing on the spatial–temporal dynamics of joint movements, this research aims to contribute a robust and efficient method for HAR, paving the way for more intuitive and less invasive activity recognition solutions.

Furthermore, the proposed system aims to address several key challenges in HAR, such as recognizing activities in naturalistic environments, handling occlusions, and differentiating between similar actions. By leveraging the graph structure, the system can maintain the spatial integrity of joint positions, even in the presence of occlusions, and distinguish between activities that may appear similar in individual frames but differ in their overall movement patterns. The integration of graph structures with deep neural networks represents a promising advancement in HAR, offering improved accuracy and applicability in real-world scenarios. This research not only aims to enhance the technical aspects of activity recognition but also to contribute to the development of intelligent systems that can seamlessly integrate into daily life, providing safety and support without compromising user comfort or privacy. By providing reliable and efficient HAR solutions, the outcomes of this research have the potential to impact various domains, including healthcare monitoring, smart home systems, and public safety.

2. Literature Review

The primary goal of HAR is to accurately identify and classify human activities based on data collected from various sensors. As mentioned before, HAR has largely relied on active or passive sensor-based methods, which involve the use of fixed or mobile sensors to imitate human movements. These methods have paved the way for understanding the complexities of human activity but come with their own set of limitations.

With the advancement of deep learning techniques, the field of HAR has witnessed a paradigm shift. Deep learning models, particularly CNNs and RNNs, have demonstrated remarkable success in learning complex feature representations from raw sensor data. These models have significantly improved the accuracy and robustness of HAR systems.

2.1. Active Sensor-Based HAR

In the early days of human activity recognition research, the field was primarily carried out through manual observation, where researchers would closely monitor and meticulously document the activities of study participants. This manual method, though direct, required significant time and effort from the researchers and was prone to subjectivity and inconsistencies inherent in human observation. The emergence of wearable sensor technologies in the late 20th century represented a pivotal moment in the evolution of this research area, enabling a more objective and quantifiable approach to activity recognition.

The introduction of devices like accelerometers, radars, lidars, and gyroscopes allowed researchers to capture detailed data on movement and orientation, providing a new lens through which to analyze human behaviors. The first sensor designed for this purpose was introduced in [1]. The authors discussed the development of a device designed to record physical activity data in an ambulatory setting. This device is capable of tracking and storing various movements throughout the day, which can then be analyzed to categorize different types of actions in daily life. The waveforms produced by the sensors contain hidden information about the character's activity. As a result, researchers often use signal-processing techniques to uncover the latent features embedded in these waveforms. The data are first preprocessed using filters and, generally, the windowing technique is used to perform feature extraction. Then, a signal transformation technique is applied for HAR systems, such as Fourier transform [2], wavelet transform [3–5], and discrete cosine transformation [6,7]. The extracted features were initially utilized with early methods that were primarily heuristic-based. These approaches involved developing rules and applying statistical methods to classify activities [8].

It is noteworthy that most often, the majority of sensor-based human activity recognition systems have been extensively applied within Internet of Things (IoT) environments [3,6,9–11]. This prevalence is driven by the growing demand for intelligent and responsive systems capable of monitoring and interpreting human activities in real time.

While sensor-based human activity recognition (HAR) systems offer numerous advantages, they also come with several limitations. Sensor accuracy and reliability can vary depending on environmental factors, sensor placement, and quality, leading to potential errors in activity detection. Battery life and energy consumption are also critical issues, particularly for wearable devices, which require regular charging and maintenance. These limitations highlight the need for careful consideration in the design and implementation of sensor-based HAR systems.

2.2. Passive Sensor-Based HAR

Unlike active sensors that require active participation or feedback from users, passive sensors gather data without explicit user input, thereby providing a more seamless and unobtrusive experience. These methods rely on sensors that monitor environmental changes without actively emitting signals, thus preserving user privacy and comfort. Passive sensors used in human activity recognition include various types, such as surveillance cameras, ambient light sensors, and barometric pressure sensors.

One common passive sensor used in HAR is the surveillance camera. Cameras capture visual data that can be analyzed to identify and classify various activities based on body movements, posture, and interactions with objects [12]. These visual data, when processed using computer vision algorithms, provide detailed insights into complex activities, such as distinguishing between different physical exercises or detecting abnormal behaviors. Moreover, HAR systems that incorporate data from cameras can revolutionize education, health monitoring, sports, and security [13].

Another type of passive sensor commonly utilized in HAR systems is the ambient light sensor. These sensors detect changes in lighting conditions within an environment, which can be indicative of certain activities. For instance, a sudden change in light levels might suggest someone entering or leaving a room or the start and end of different tasks based on lighting patterns. These types of data can complement visual data from cameras, providing a more comprehensive understanding of the context in which activities occur [14,15].

Once the sensor data are acquired, it is necessary to adopt an appropriate approach to analyze the data in order to extract the latent information and to identify the action correctly. Deep learning methods are among the most widely used approaches, having established

themselves as a prevailing direction in machine learning, outcompeting conventional methods in several computer vision tasks. Deep learning algorithms can extract features automatically from raw data, removing the reliance on handcrafted feature detectors and descriptors. Among them, convolutional neural networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers are some of the most widely used deep learning techniques for HAR systems.

Ref. [16] proposed an optical flow-based two-stream CNN that overcomes the computational inefficiency of optical flow. It substitutes optical flow with motion vectors, which are computationally less expensive to compute and can be directly obtained from the video stream. In an effort to alleviate the quality deterioration brought about by motion vectors, the authors learned deeply transferred Motion Vector CNNs (DTMV-CNNs), employing a knowledge transfer strategy from optical flow CNNs. It maintains competitive accuracy and considerably accelerates the speed with frame rates over 390 fps, which is great for real-time circumstances. The framework was evaluated on standard datasets like UCF101, HMDB51, and THUMOS14, and achieved state-of-the-art accuracy and efficiency.

Ref. [17] introduced ActionXPose, a new pose-level HAR algorithm capable of performing real-time HAR in everyday scenarios by CCTV-like cameras. The approach utilizes 2D poses extracted using OpenPose to classify human actions while addressing challenges such as occlusions and missing data. ActionXPose extracts both low- and high-level features from body poses, which are processed using a Long Short-Term Memory Neural Network (LSTM) and a 1D convolutional neural network (CNN) for classification. The study also introduced a new dataset, ISLD, specifically designed for realistic pose-level HAR, and demonstrated the robustness of ActionXPose through extensive experiments settings. The method achieves state-of-the-art performance, showing high accuracy and robustness across various datasets.

Ref. [18] designed a 3D convolutional neural network (3DCNN) human activity recognition (HAR) framework based on video. In contrast to classical 2D CNNs, which only consider spatial information, the 3DCNN takes advantages of temporal characteristics of input video sequences and generates a volumetric 3D activation map including spatiotemporal information. The optimal architecture includes several types of layers: 3D convolutions, MaxPooling3D, batch normalization, and fully connected layers, resulting in high precision and robustness. On benchmarks like UCF YouTube Action and UCF101, their test accuracies reached a whopping 85.2% and 79.9%. This model surpassed prior motion-, static-, and hybrid-based architectures.

In [19], the authors proposed a computationally efficient HAR method based on skeleton data generated by OpenPose. The combined CNN and LSTM Network presented an efficient model that only requires skeleton data and does not need to convert the data into RGB images; it achieved state-of-the-art accuracy of 94.4% on the MCAD dataset and 91.67% on the IXMAS dataset. This approach has practical applications in fields such as video surveillance, human–computer interaction, and healthcare.

Ref. [20] proposed a unified framework of CNNs and Vision Transformers (ViTs) for HAR. The system is based on the sequence of the MoveNet to extract the spatial features of the image and then shape the next convolution image features with the help of ResNet-18, EfficientNet, and other pre-trained models. The class learns spatiotemporal dependencies and achieves 87.50% and 83.41% accuracy on UCF 101 and UCF 50, respectively. The study showed that CNNs are usually great at improving model robustness and efficiency, while Transformers can address some challenges associated with the actions like action ambiguity and misclassification of similar motions.

Ref. [21] addressed occlusion in HAR, which is often perceptively omitted in the literature conducted in ideal conditions. They introduced a technique that replicates

occlusions by dropping the skeletal joints for specific body parts and trains a CNN using both completely visible and occluded data samples. Utilizing datasets like PKU-MMD and NTU-RGB+D, their results demonstrated how occluded samples in training significantly improved recognition accuracy. Their research highlights the need to address real-life challenges, such as occlusion, in developing more resilient HAR systems.

Despite the advantages of passive sensor-based HAR systems, several challenges remain. One major concern is the variability in data quality due to environmental factors, such as lighting changes, weather conditions, and obstructions that can affect sensor readings. For example, the use of cameras raises significant privacy concerns and requires substantial data storage and computational resources to process the visual information efficiently [22]. As a result, future research in passive sensor-based HAR should focus on the ethical implications of using passive sensors, particularly regarding privacy concerns and the potential for misuse in surveillance [23].

3. Proposed Methodology

The proposed human activity recognition (HAR) system, as illustrated in Figure 1, is structured to process data from input to output through a series of specialized components. The process begins with a surveillance camera that captures video footage from public spaces. This video is then processed frame-by-frame in the frame preprocessing stage, where each frame is enhanced to suppress distortions and improve image clarity. Following preprocessing, the system uses pose estimation techniques to detect human joints, identifying key points on the body such as the head, shoulders, and knees. These joints are then tracked across consecutive frames to monitor their positions over time, which is essential for analyzing the dynamics of human activities.

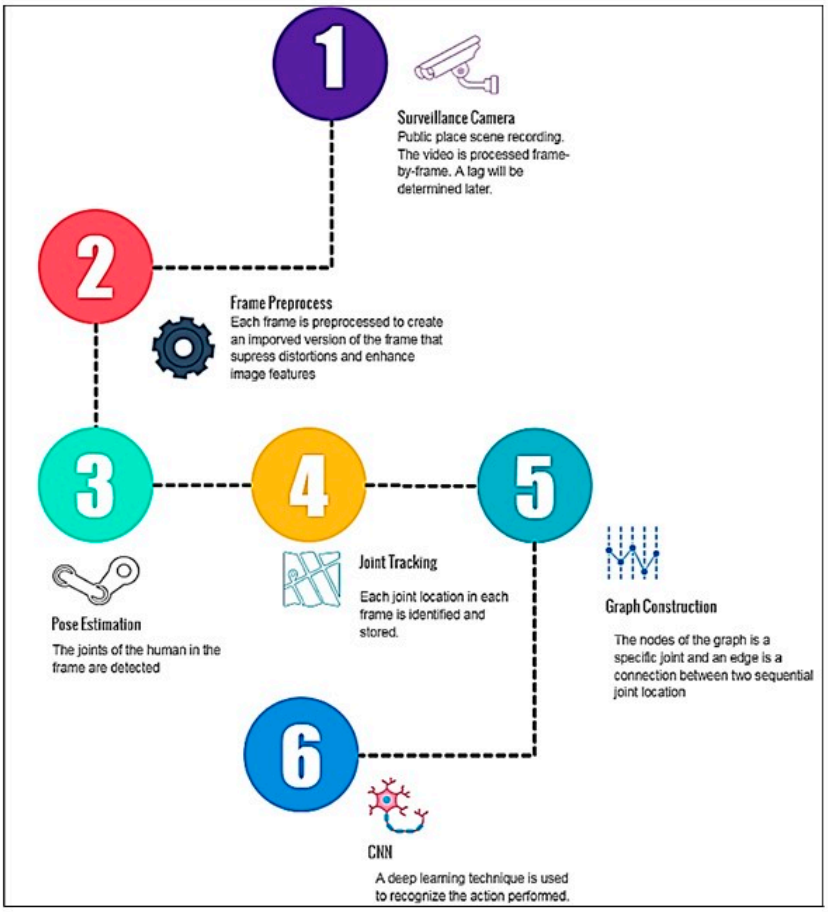

Figure 1. System architecture of the proposed methodology.

Subsequently, the system constructs a graph from the detected joints in each frame, where a node represents a specific joint and edges denote the connections between the same joint in two consecutive frames. This graph-based representation encapsulates both spatial relationships and temporal dynamics, offering a detailed understanding of human movement. Finally, a convolutional neural network (CNN) is employed to analyze these graphs. The CNN, trained on various patterns in the graph data, classifies the observed activities by interpreting the spatial–temporal dynamics inherent in the joint movements. This architecture is designed to efficiently and accurately recognize human activities, making it applicable in fields like public safety, healthcare monitoring, and smart surveillance. By integrating graph structures with deep learning techniques, the system aims to deliver robust and reliable activity recognition, even in complex and dynamic environments.

3.1. Camera Raw Data

The HAR system initiates its process with a surveillance camera that captures video footage from public spaces, providing continuous video streams as the primary input data. For HAR systems, there are mainly two types of datasets: sensor-based and vision-based datasets. The methodology applied in this research considers vision-based datasets only. Vision-based datasets exist for two types of actions that are static and dynamic actions. In static-actions datasets, a description of the action using orientation and limb position in space can be found, while dynamic action datasets are related to videos that describe the movement of static activities [24].

Accurate recognition of human actions in videos is challenging due to variations in background, lighting, and the dynamic nature of human movements. To address these challenges and robustly evaluate the performance of action recognition algorithms, leveraging multiple datasets is essential. This research aims to measure the performance of our proposed algorithm by using similar actions from two well-known datasets: UCF101 [25] and Kinetics-400 [26]. By focusing on common actions across these datasets, we can ensure a fair comparison and comprehensive assessment of our algorithm's capabilities as well as enhance the generalizability of the recognition model.

The UCF101 dataset consists of 101 action categories, encompassing 13,320 video clips, mainly sourced from YouTube. It provides a rich variety of sports and everyday activities captured in different environments. The Kinetics dataset, particularly in its Kinetics-400 version, contains 400 action categories and approximately 306,245 video clips from YouTube, offering an extensive collection of action scenarios. These datasets are widely used benchmarks in the field, allowing for rigorous testing and validation of action recognition models.

As illustrated in Table 1, we identified 59 common actions across the UCF101 and Kinetics datasets with a total of 63,890 videos. The number of videos used from the UCF101 dataset was 7864, representing approximately 59% of the dataset, while 56,026 videos were used from the Kinetics-400 dataset, representing approximately 18.3% of the dataset. The selection of 59 common actions between the UCF-101 and Kinetics-400 datasets ensures a consistent basis for evaluating the proposed HAR system's performance across different settings and environments. This approach allows for a comprehensive examination of the system's ability to generalize across datasets and recognize activities with inherent variability.

Table 1. List of similar actions across the UCF101 and Kinetics datasets used to evaluate the performance of the proposed action recognition algorithm.

Similar Actions 1	**Similar Actions 2**	**Similar Actions 3**	**Similar Actions 4**
Archery (1292)	Cutting In Kitchen (1589)	Knitting (814)	Salsa Spins (1281)
Baby Crawling (1282)	Diving (1118)	Long Jump (962)	Shaving Beard (1142)
Band Marching (1301)	Drumming (1069)	Lunges (886)	Shotput (1131)
Basketball (1188)	Field Hockey Penalty (1043)	Mopping Floor (716)	Skateboarding (1259)
Basketball Dunk (1236)	Floor Gymnastics (1268)	Playing Guitar (1295)	Skiing (612)
Bench Press (1266)	Frisbee Catch (1186)	Playing Piano (796)	Skijet (1240)
Biking (1186)	Golf Swing (975)	Playing Violin (1242)	Sky Diving (615)
Blowing Candles (1259)	Haircut (788)	Playing Cello (1245)	Soccer Juggling (631)
Body Weight Squats (1260)	Hammer Throw (1298)	Playing Flute (630)	Surfing (877)
Bowling (1233)	High Jump (1077)	Pole Vault (1133)	Tai Chi (1170)
Boxing Punching Bag (646)	Horse Riding (1295)	Pull Ups (1221)	Throw Discus (1234)
Breaststroke (934)	Hula Hoop (1254)	Punch (1310)	Trampoline Jumping (809)
Brushing Teeth (1280)	Javelin Throw (1029)	Push Ups (716)	Volleyball Spiking (920)
Clean and Jerk (1014)	Juggling Balls (1044)	Rock Climbing Indoor (1275)	Walking With A Dog (1268)
Cliff Diving (1231)	Kayaking (1287)	Rope Climbing (532)	

The decision to use only the 59 common actions shared between the UCF-101 and Kinetics-400 datasets, despite the fact that each dataset contains a greater number of actions, was made to ensure consistency and comparability in the evaluation. Using all the actions from both datasets would have introduced variability and imbalance, as each dataset has many unique activities that are not present in the other. This inconsistency could lead to biases when comparing the model's performance, making it difficult to establish a fair benchmark. By focusing solely on the common actions, this study maintains a standardized and uniform evaluation framework that allows for a more direct comparison of model effectiveness.

Additionally, limiting the number of actions helps mitigate the risk of overfitting. When training on an excessive and potentially unbalanced set of activities, the model may learn specific patterns that do not generalize well beyond the dataset. Such overfitting is particularly likely when dealing with niche or highly specific actions that are over-represented in one dataset but absent in the other. By focusing on the common actions, the model is trained on a balanced set of movements that are more likely to generalize well, enhancing the applicability of the HAR system to a broader range of real-world situations.

In addition, in order to report results for comparison with other state-of-the-art methods, the proposed approach was also tested on the full UCF-101 dataset to examine the generalization and scalability of the proposed method. This expanded perspective sheds

light on the overall performance of the model across a wide spectrum of actions and highlights its potential in discussing activity recognition in realistic scenarios.

3.2. Frame Preprocessing

Given the diverse range of videos in the dataset used for this research, each frame must undergo preprocessing to standardize the visual quality. This involves adjusting illumination and contrast to ensure consistent frame quality. Preprocessing not only enhances frame quality but also reduces training time and improves model accuracy. In this research, several preprocessing techniques are applied, including mean normalization, histogram equalization, and data rescaling. These techniques help to normalize the data and enhance the overall effectiveness of the HAR system.

3.2.1. Mean Normalization

Mean normalization is a preprocessing technique used to standardize data by adjusting their scale and distribution. The equation for mean normalization is Equation (1).

$$F(x) = (x - \mu)/\mathcal{R} \tag{1}$$

where x represents the original data point, μ is the mean of the dataset, and $\mathcal{R}$ is the range, calculated as the difference between the maximum and minimum values in the dataset.

This technique is particularly useful in reducing the influence of outliers and ensuring that the data are centered around zero, which can enhance the performance of machine learning models by improving convergence during training.

3.2.2. Histogram Equalization

Histogram equalization is a technique used to enhance the contrast of an image by redistributing its intensity values. The process is mathematically represented by Equation (2).

$$S_k = \left(\frac{L-1}{MN}\right)\sum_{i=0}^{k} n_i \tag{2}$$

where S_k denotes the new intensity value for the k-th pixel in the equalized image. In this equation, L represents the total number of possible intensity levels, M and N are the dimensions of the image, indicating the total number of pixels, and n_i is the number of pixels with the intensity value i in the original image. The term $\sum_{i=0}^{k} n_i$ calculates the cumulative distribution function (CDF) up to intensity level k. By adjusting the pixel values based on the CDF, histogram equalization spreads out the intensity values over the available range, thus enhancing the contrast of the image. This technique is especially useful for images with poor contrast, allowing more details to be visible by broadening the range of pixel intensities.

3.2.3. Data Rescaling

Data rescaling, commonly referred to as min–max normalization, is a method used to adjust the range of data values to a specific scale, typically between 0 and 1 or -1 and 1. This technique is represented by Equation (3).

$$x' = (x - x_{min})/(x_{max} - x_{min}) \tag{3}$$

where x' is the normalized value, x is the original data value, and x_{min} and x_{max} are the minimum and maximum values of the dataset, respectively. By applying this formula, the data are scaled so that the minimum value becomes 0 and the maximum value becomes 1. Min–max normalization is particularly useful in machine learning as it ensures that

different features contribute proportionately to the model's training process, preventing features with larger scales from disproportionately influencing the model's outcomes.

3.3. Pose Estimation

In this research, the OpenPose algorithm was employed for pose estimation, a crucial step in understanding and analyzing human activities. OpenPose is a state-of-the-art algorithm designed to detect human poses from images and videos by identifying key points on the human body, such as the head, shoulders, elbows, wrists, hips, knees, and ankles. The algorithm provides a detailed skeleton-like representation for each individual in the scene, capturing the spatial relationships between joints. This capability makes OpenPose particularly suitable for complex scenarios involving multiple individuals or dynamic movements.

The process begins by feeding video frames into the OpenPose system, where each frame is processed independently. OpenPose uses a two-branch multi-stage CNN architecture. The first branch predicts confidence maps for the location of each key point, while the second branch predicts part affinity fields—vector fields that encode the location and orientation of limbs. These outputs are combined to construct a coherent representation of the pose for each person in the image. The system iteratively refines its predictions through several stages, enhancing the accuracy of joint detection and association. This iterative refinement is crucial for ensuring high precision, especially in environments where joints are partially occluded.

OpenPose is known for its accuracy and robustness, performing well on multiple people with occlusions. Its iterative multi-stage convolutional neural network architecture progressively fine-tunes pose predictions, facilitating very efficient performance on challenging HAR problems in highly populated scenes. On the other hand, Mediapipe performs very well in real time and requires fewer computations; therefore, it is better suited for mobile and embedded systems. OpenPose is highly validated through academic research topics and offers better multi-person support and more integration flexibility, but it is much more complex to set up and requires high-performance hardware. Mediapipe, on the other hand, aims for simplicity and speed in the deployment, providing fast pose estimation of a single person without state-of-the-art accuracy—known as Mediapipe Pose—primarily for cases where other people may block the view. OpenPose performs well for research-oriented HAR or where a detailed multi-person analysis of its output is needed as a preprocessing step to model input, while Mediapipe is more suitable for real-time and resource-constrained tasks.

By applying OpenPose, this research achieved a high level of accuracy in pose estimation, which was essential for subsequent stages of analysis. The detailed representation of joint positions and the relationships between them enabled a comprehensive understanding of human movement patterns. These data formed the basis for constructing the graph-based representations used in this study to analyze and classify activities. The robustness of OpenPose in handling varying scales, orientations, and occlusions makes it a reliable tool for real-world applications, ranging from healthcare monitoring and sports analysis to security and surveillance. The algorithm's ability to work with standard video inputs without the need for specialized equipment further enhances its practicality and accessibility for various research and applications.

3.4. Joint Tracking

Capturing the temporal information in a video is very challenging. However, this can be obtained by tracking joints. The objective of joint tracking is to continuously monitor and analyze human body movements throughout the video frames. This process involves

identifying and following the positions of key body joints—such as the head, shoulders, elbows, hips, knees, and ankles—across consecutive frames in a video sequence. By accurately tracking these joints, the system can map the trajectory of each joint over time, capturing the dynamic aspects of human movement. This temporal information is essential for understanding activities, as it allows the system to differentiate between various actions based on the patterns and sequences of joint movements. Joint tracking also plays a pivotal role in handling occlusions and other visual challenges as it maintains the continuity of motion data even when some joints are temporarily not visible. This capability is crucial for applications in sports analysis, healthcare monitoring, and surveillance, where precise and reliable tracking of body movements is necessary.

In Figure 2, the coordinates of the right wrist and nose of a person waving their right hand in a video are illustrated over time. The top left subplot depicts the X coordinate of the right wrist, featuring both the original and smoothed data series. Similarly, the top right subplot shows the Y coordinate of the right wrist. The bottom left and right subplots display the X and Y coordinates of the nose, respectively, also with both original and smoothed data series.

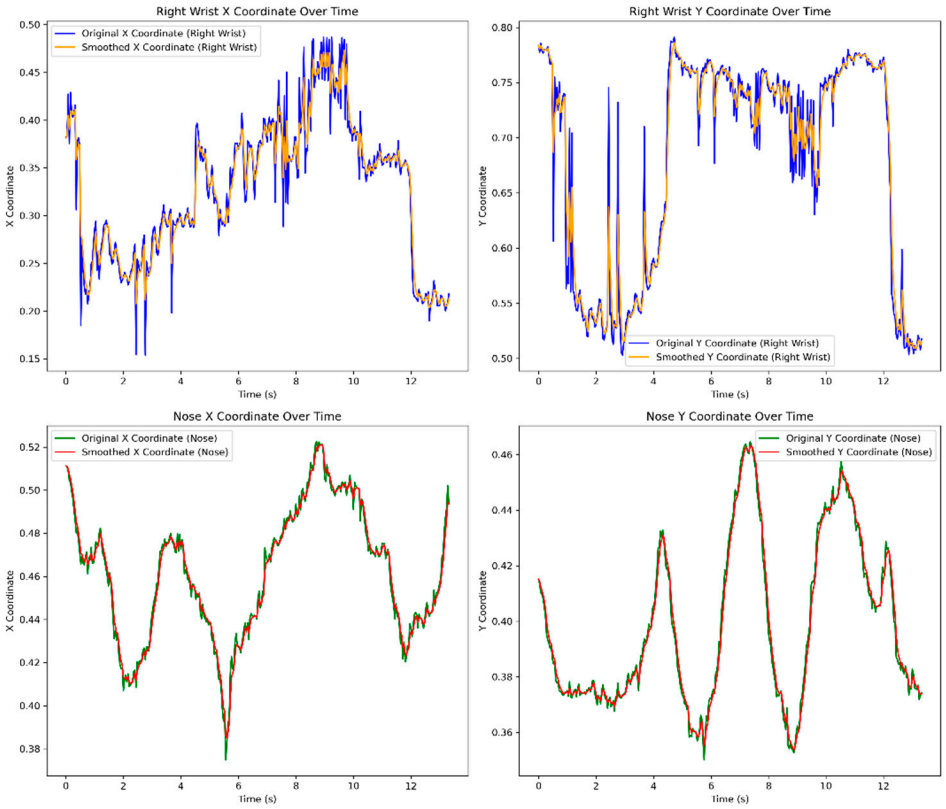

Figure 2. Temporal evolution of right wrist and nose coordinates during a waving motion.

The original data, depicted in the top subplots of Figure 2, exhibit noticeable variability, indicating potential noise or rapid movements. In contrast, the smoothed version offers a clearer depiction of the overall movement patterns by reducing the influence of noise. This smoothing effect is particularly evident in the right wrist coordinates, where fluctuations are significantly attenuated.

The visualization effectively communicates the temporal evolution of these coordinates, facilitating the analysis of movement patterns. The distinction between the original and smoothed data underscores the importance of data processing in motion analysis, highlighting the utility of smoothing techniques in revealing underlying trends amidst noisy data.

3.5. Graph Construction

Graph construction is the heart of the proposed HAR system as it involves transforming the spatial and temporal data of human joints into a structured format suitable for deep learning analysis. In this process, each human joint detected in the pose estimation phase is represented as a node in the graph. The connections between these nodes, known as edges, represent the anatomical connections between the joints. These edges capture both the spatial proximity and the sequential movement patterns of the joints throughout the video, providing a comprehensive view of the body's configuration and dynamics.

The graph construction process begins with the definition of nodes and edges. Each node corresponds to a specific joint, such as the head, shoulders, elbows, wrists, hips, knees, or ankles. The edges between nodes are defined based on the natural anatomical connections (e.g., shoulder to elbow, elbow to wrist) and are weighted to reflect the significance or strength of these connections. The weights determine the relative velocity between two joints in the video.

To construct the graph, the joint positions are first normalized to a consistent scale to account for variations in camera angles, distances, and subject sizes. This normalization ensures that the graph accurately represents the relative positions and movements of the joints, regardless of external factors. The nodes and edges are then instantiated, with edges being annotated with weights corresponding to the relative velocity between two joints to capture temporal dynamics between them during the activity.

To calculate the relative velocity between the joints, we applied Equation (4), as follow:

$$v_{relative} = v_{joint\_i} - v_{joint\_j} \tag{4}$$

where v_{joint_i} and v_{joint_j} are the velocity vectors of joint i and joint j. The output is a vector that represents how fast and in what direction *joint_i* moves relative to *joint_j* in the video. The velocity vector of a joint is calculated by deriving the first derivative of the position data over time, as in Equation (5). This provides insights into the speed and movement direction of the joint for all video frames.

$$v(t) = \frac{dp(t)}{dt} = \left(\frac{dx(t)}{dt}, \frac{dy(t)}{dt}\right) \tag{5}$$

where $p(t) = (x(t), y(t))$ is the position vector as a function of time t. After finding the relative velocity between two joints, we find the relative velocity magnitude and represent this value as the edge weight between the incorporated joints. The calculation of relative velocity magnitude is found using Equation (6) as the function $M(v_{relative})$.

For two joints with positions (x_1, y_1) and (x_2, y_2) at time t, and their positions (x_1', y_1') and (x_2', y_2') at time $t + \Delta t$, the relative velocity magnitude can be calculated as

$$M(v_{relative}) = \frac{\sqrt{\left((x_1' - x_1) - (x_2' - x_2)\right)^2 + \left((y_1' - y_1) - (y_2' - y_2)\right)^2}}{\Delta t} \tag{6}$$

This calculation shows how the relative position between the two joints changes over the time interval Δt, providing a measure of their relative motion.

The resulting graph encapsulates both the spatial configuration and temporal evolution of the body, allowing for robust recognition and classification of human activities based on complex movement patterns. By utilizing this structured representation, the system can better handle variations in human poses and movements, improving accuracy and robustness in real-world applications. It is noteworthy to mention that in the case of occlusions, the magnitude between an existing joint and the occluded joint is set to zero.

Figure 3 illustrates the graph structure as a heatmap, visualizing the average relative velocities between different joint pairs during a human waving their right hand. Using a color gradient from dark red to yellow, where dark red indicates lower relative velocities and yellow indicates higher relative velocities, the heatmap offers a clear depiction of joint activity. The diagonal elements are black, reflecting that the relative velocity of a joint with itself is zero. Notably, the brightest areas (yellow) highlight higher relative velocities between the right shoulder, right elbow, and right wrist, consistent with the waving motion. Conversely, the lower body joints display darker colors, indicating minimal movement. An asymmetry is evident, with higher velocities on the right side of the body, aligning with the right-hand waving action. This heatmap effectively showcases the dynamic coordination between body joints during the activity, providing valuable insights into joint movement and synchronization.

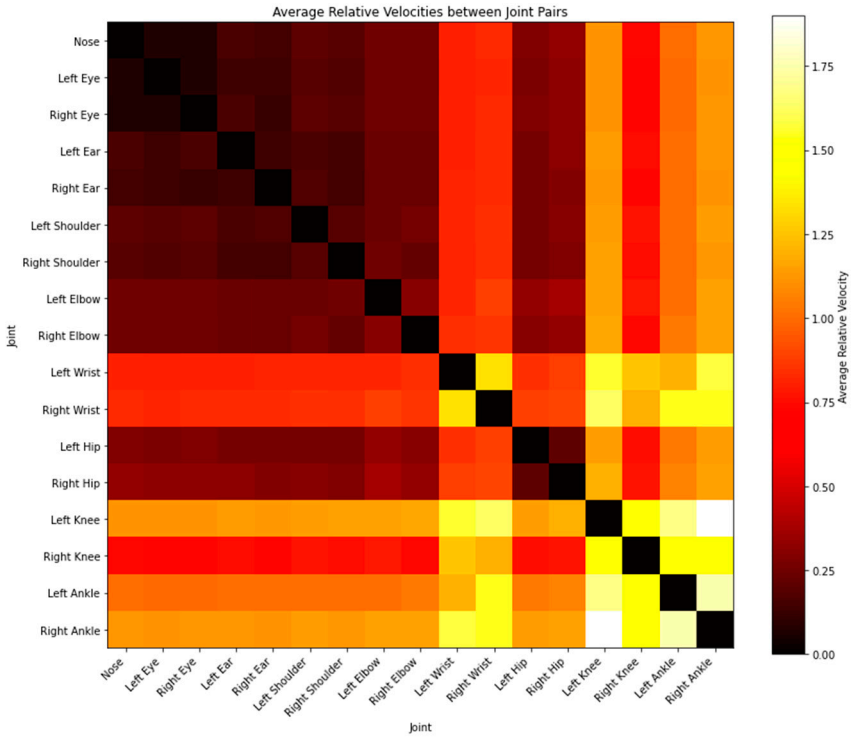

Figure 3. Heatmap of average relative velocities between joint pairs during right-hand waving motion.

Although pose-based data typically lend themselves to capturing spatial and temporal relationships simultaneously, graph representations are perfect for preserving the non-Euclidean topology of data. Different from Transformers with self-attention mechanisms or Temporal Convolutional Networks (TCNs) with fixed temporal scopes, graphs encode joint connectivity and temporal hierarchies in skeletal movements, allowing interpretable and effective modeling of human activities. Moreover, they allow for a concise and regularized representation of pose data with lower computational load (in contrast to the large feature maps created from Transformers or the sequential progression of TCNs) since location landmarks provide a minimal description of the human body compared to different architectures used. Such efficiency renders graph-based methods exceptionally applicable to real-time tasks.

Graphs naturally only represent the joints present in the data and their interconnections, making them immune to missing or occluded data, whereas Transformers and TCNs might require different imputation or padding strategies that may introduce noise and potentially reduce the accuracies in harder scenarios.

3.6. Convolution Neural Network

The classification model used in this research starts with an input layer designed to receive 17 × 17 (17 joints considered in this research) matrix with a single channel that represents the graph structure describe in Section 3.5. Figure 4 illustrates the architecture of the CNN model. The input layer feeds into a series of convolutional layers, each followed by a max pooling layer. Equation (7) provides the mathematical formula of a convolution layer, and Equation (8) provides the mathematical formula of the max pooling layer. The first convolutional layer consists of 32 filters with a kernel size of 3 × 3 and uses the ReLU activation function, which introduces non-linearity by outputting the input directly if it is positive and zero otherwise. This layer is followed by a max pooling layer with a 2 × 2 pool size, which reduces the spatial dimensions of the feature maps by taking the maximum value over each 2 × 2 block, thus reducing the number of parameters and computation in the network.

$$(I * K)(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} I(i+m, j+n) \cdot K(m,n) \tag{7}$$

where I is the input matrix, K is the convolution kernel, and M and N are the dimensions of the kernel.

$$P(i,j) = \max\{I(i+2,\ j+2)\} \tag{8}$$

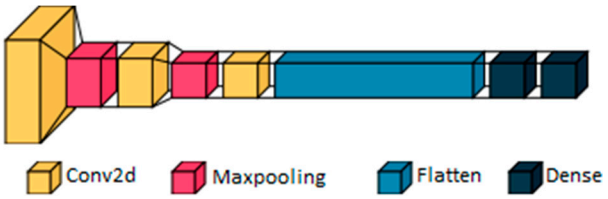

Figure 4. Architecture of the CNN model.

Next, a second convolutional layer with 64 filters, a 3 × 3 kernel, and ReLU activation is added. This layer employs "same" padding to ensure that the output feature map has the same spatial dimensions as the input. Another max pooling layer with a 2 × 2 pool size follows, further reducing the spatial dimensions of the feature maps. The third convolutional layer comprises 128 filters, also with a 3 × 3 kernel size and ReLU activation, and uses "same" padding to maintain the spatial dimensions of the feature maps.

Following the convolutional layers, the feature maps are flattened into a one-dimensional vector, which serves as the input to the fully connected (dense) layers. The first dense layer contains 128 neurons and uses the ReLU activation function to introduce non-linearity. The second dense layer also has a number of neurons (optimized during hyperparameter tuning) and uses the ReLU activation function. The final dense layer is the output layer, containing 59 neurons (corresponding to the number of classes in the dataset), and uses the softmax activation function. The softmax function converts the output scores into probabilities, facilitating the classification of the input images into one of the five classes.

3.7. Hyperparameter Optimization Using the Firefly Algorithm

The CNN model described in Section 3.6 is a traditional architecture with three convolutional layers, where the number of filters increases in each subsequent layer. Specifically, the first convolutional layer uses 32 filters, the second uses 64 filters, and the third uses 128 filters. These filter counts are commonly used in simple CNN models to progressively capture more complex features from the input data, and each layer refines the features extracted by the previous layers.

The accuracy of such a model is significantly influenced by the chosen hyperparameters, including the number of filters in each convolutional layer, learning rate, number and size of dense layers, batch size, and the number of neurons in the dense layers. To enhance the performance of the CNN model in this research, hyperparameters were optimized using the Firefly Optimization Algorithm [27].

The Firefly Optimization Algorithm is a metaheuristic optimization technique that simulates the behavior of fireflies, where the attractiveness and movement of fireflies are governed by their brightness. The key idea behind the Firefly Algorithm is that each firefly is attracted to brighter fireflies, with the brightness being proportional to the objective function being optimized.

In this research, the Firefly Algorithm was employed to optimize the hyperparameters of the CNN model, including the number of neurons in the dense layers, learning rate, batch size, and number of filters in the convolutional layers. The optimization process involved using 10 fireflies over 20 iterations, allowing for effective exploration of the hyperparameter space to find optimal values. The steps of the Firefly Algorithm are as follows:

1. Initialization.

Initialize a population of fireflies with random positions in the search space. Each firefly represents a potential solution. In this research, the initial number of fireflies was 10, with 20 iterations.

2. Attractiveness.

The light intensity I of a firefly at a particular location x is determined by the objective function $f(x)$. The attractiveness β of a firefly is given by Equation (9):

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{9}$$

where β_0 is the maximum attractiveness, γ is the light absorption coefficient, and r is the distance between two fireflies. In this research, γ was set to 1.0.

3. Distance Calculation.

The distance r_{ij} between two fireflies i and j at positions x_i and x_j is calculated using the Euclidean distance, as in Equation (10):

$$r_{ij} = \left\| x_i - x_j \right\| \tag{10}$$

4. Movement.

A firefly i moves towards a more attractive (brighter) firefly j. The movement is determined by Equation (11):

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} \left(x_j - x_i \right) + \alpha (rand - 0.5) \tag{11}$$

where α is a randomized parameter and is $rand$ a random number uniformly distributed between 0 and 1.

The value of β_0 used in this research was set to 0.2, and the value of α was 0.5. The term $\beta_0 e^{-\gamma r_{ij}^2} \left(x_j - x_i \right)$ ensures that fireflies move towards each other based on their attractiveness, which is stronger for closer and brighter fireflies. The exponential factor $e^{-\gamma r_{ij}^2}$ ensures that the influence decreases with distance, making distant fireflies less attractive. The term $\alpha (rand - 0.5)$ introduces stochastic behavior, preventing the algorithm from getting stuck in local optima and enhancing the exploration of the search space.

3.8. Evaluation Metrics

To assess the performance of the proposed human activity recognition (HAR) system, the following evaluation metrics were used:

Accuracy: Measures the proportion of correctly classified samples to the total number of samples:

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

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Precision: Evaluates the proportion of correctly predicted positive samples out of all predicted positive samples:

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity): Measures the ability of the model to correctly identify all actual positive samples:

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: Provides a balance between precision and recall by computing their harmonic mean:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

These metrics were chosen to provide a comprehensive evaluation of the model's performance, addressing both classification accuracy and the trade-offs between precision and recall.

The importance of using these metrics to validate machine learning models was emphasized in [28].

4. Results

In this section, we present the outcomes of our experiments, demonstrating the effectiveness of our proposed human activity recognition (HAR) system that integrates graph structures and deep neural networks. The evaluation is conducted in comparison with traditional HAR methods, with a focus on several performance metrics: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive view of the algorithm's ability to correctly identify human activities while minimizing false detections.

4.1. Experimental Setup

To evaluate the performance of the proposed HAR system, we utilized two well-known datasets: UCF-101 and Kinetics-400. We selected 59 common actions shared between these datasets to ensure consistency in evaluating our system's ability to generalize across different environments and contexts. The use of only common actions helped maintain a balanced training dataset, reducing the risk of overfitting and ensuring a more representative assessment of the model's capabilities.

The experiments were conducted in a controlled environment using a standard computational setup. We applied preprocessing techniques including mean normalization, histogram equalization, and data rescaling to enhance the quality of video frames and facilitate the efficient training of the deep learning model. The OpenPose algorithm was used for pose estimation, allowing us to detect and track the 3D trajectories of human joints, which were then represented as graph structures.

To facilitate a more detailed analysis of the results, the 59 selected actions were categorized into four groups based on the movement speed of human joints during the actions: Sports and Athletics, Household and Routine Activities, Musical and Performing

Arts, and Outdoor and Adventure Activities. Table 2 illustrates the actions of each group. It is important to note that the model's final layer consists of 59 outputs, with each output corresponding to a specific action. To determine the accuracy for each group, we calculated the average accuracy of all actions within that group.

Table 2. Categorization of actions with corresponding total video counts across the four defined categories.

Category	**Total No. Videos**	**Actions**
Household and Routine Activities	8870	Blowing Candles, Brushing Teeth, Cutting In Kitchen, Knitting, Mopping Floor, Shaving Beard, Baby Crawling, Haircut
Musical and Performing Arts	10,127	Band Marching, Drumming, Floor Gymnastics, Playing Cello, Playing Flute, Playing Guitar, Playing Piano, Playing Violin, Salsa Spins
Outdoor and Adventure	15,477	Biking, Kayaking, Rope Climbing, Skijet, Walking With A Dog, Sky Diving, Frisbee Catch, Skate Boarding, Skiing, Surfing, Tai Chi, Cliff Diving, Hula Hoop, Push Ups, Juggling Balls
Sports and Athletics	29,416	Archery, Basketball, Basketball Dunk, Bench Press, Body Weight Squats, Bowling, Boxing Punching Bag, Diving, Field Hockey Penalty, Golf Swing, Hammer Throw, High Jump, Horse Riding, Javelin Throw, Long Jump, Lunges, Pole Vault, Pull Ups, Shotput, Throw Discus, Trampoline Jumping, Volleyball Spiking, Punch, Rock Climbing Indoor, Breaststroke, Clean and Jerk, Soccer Juggling

4.2. Hyperparameter Optimization Using the Firefly Algorithm

After designing the initial architecture of the HAR system, the Firefly Optimization Algorithm, as described in Section 3.7, was applied to fine-tune the model's hyperparameters. The primary objective of this optimization was to enhance the model's overall performance by determining the most effective combination of hyperparameters, such as the learning rate, batch size, and number of neurons in the dense layers, which are known to have a significant impact on the model's ability to generalize and accurately classify human activities.

The results before and after optimization are summarized in Table 3. The application of the Firefly Algorithm resulted in a significant improvement in all key performance metrics. The overall accuracy of the system increased from 85.1% to 88.9%, precision improved from 83.3% to 86.4%, recall increased from 84.6% to 87.0%, and the F1-score rose from 83.5% to 86.7%. This demonstrates the effectiveness of the Firefly Optimization Algorithm in fine-tuning the model to achieve higher classification accuracy while maintaining a balanced trade-off between precision and recall.

Table 3. Hyperparameters before and after firefly optimization.

Hyperparameter	***Initial Value***	***Optimized Value***
Learning Rate	0.001	0.0005
Batch Size	32	128
Kernel Size	3×3	5×5
Number of Filters (Conv1)	32	46
Number of Filters (Conv2)	64	96
Number of Filters (Conv3)	128	148
Number of Neurons (Dense1)	128	512
Number of Neurons (Dense2)	128	256

The hyperparameters tuned during the optimization process are listed in Table 4. The most significant changes were seen in the learning rate, the number of filters in the convolutional layers, and the number of neurons in dense layers. The optimized model with these tuned hyperparameters demonstrated improved learning efficiency and convergence, as reflected in the final performance metrics.

Table 4. Comparison of model performance before and after optimization.

Metric	*Before Optimization*	*After Optimization*
Accuracy	85.1%	88.9%
Precision	83.3%	86.4%
Recall	84.6%	87.0%
F1-Score	83.5%	86.7%

The training and validation before and after optimization are depicted in Figure 5. Before optimization, the training accuracy steadily increased and plateaued at around 85.1%, indicating that the model was learning effectively but had reached its performance limit. The validation accuracy, however, reached only about 81.7%, suggesting that the model's performance on unseen data was weaker compared to the training data. The noticeable gap between training and validation accuracies before optimization indicates that the model was prone to overfitting and had limited generalization capability.

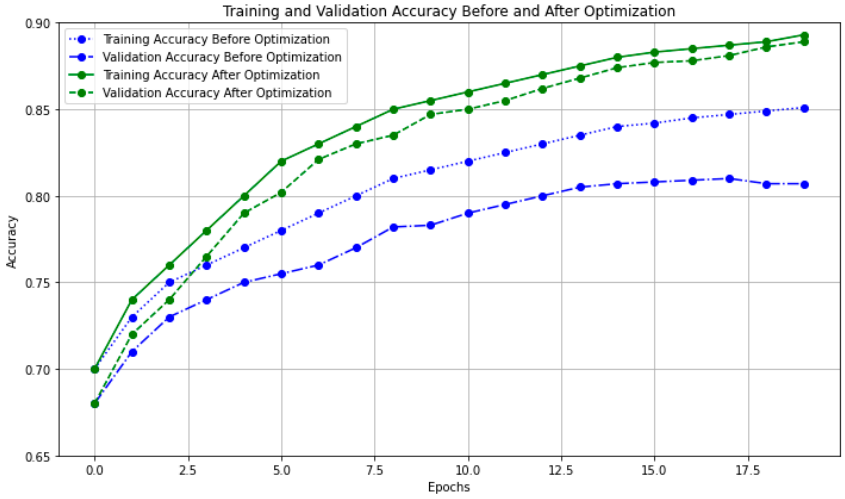

Figure 5. Training and validation accuracy before and after optimization.

After optimization, the training accuracy improved more rapidly, reaching a final accuracy of 88.9%, demonstrating the significant impact of the optimization process. Similarly, the validation accuracy saw a substantial improvement, rising to 88.8%, closely matching the training accuracy. This indicates that the optimization not only enhanced the model's training performance but also significantly improved its generalization ability. The optimization process successfully boosted both training and validation accuracy, making the model more robust and less prone to overfitting.

4.3. Overall System Performance

The optimized system's overall performance was measured using four key metrics: accuracy, precision, recall, and F1-score. As illustrated in Figure 6, the system achieved an overall accuracy of 88.9% across all actions, demonstrating its effectiveness in recognizing a wide range of human activities.

The precision, which measures the proportion of correctly identified actions out of all predicted actions, was recorded at 86.4%. This indicates that the system was able to avoid a significant number of false positives. The recall, which indicates the system's ability to correctly identify all actual instances of an action, was 87.0%. The F1-score, a harmonic mean of precision and recall, was 86.7%, showing a balanced performance between correctly identifying activities and minimizing false detections and highlighting the system's overall reliability in dealing with both complex and simpler activities.

4.4. Performance by Action Category

As mentioned before, we evaluated the system's performance across four distinct action categories: Household and Routine Activities, Musical and Performing Arts, Outdoor and Adventure Activities, and Sports and Athletics. The actions in these categories differ significantly in terms of joint dynamics, complexity, and movement patterns, making them ideal for testing the versatility of the system. The system's performance in each category is summarized in Figure 6.

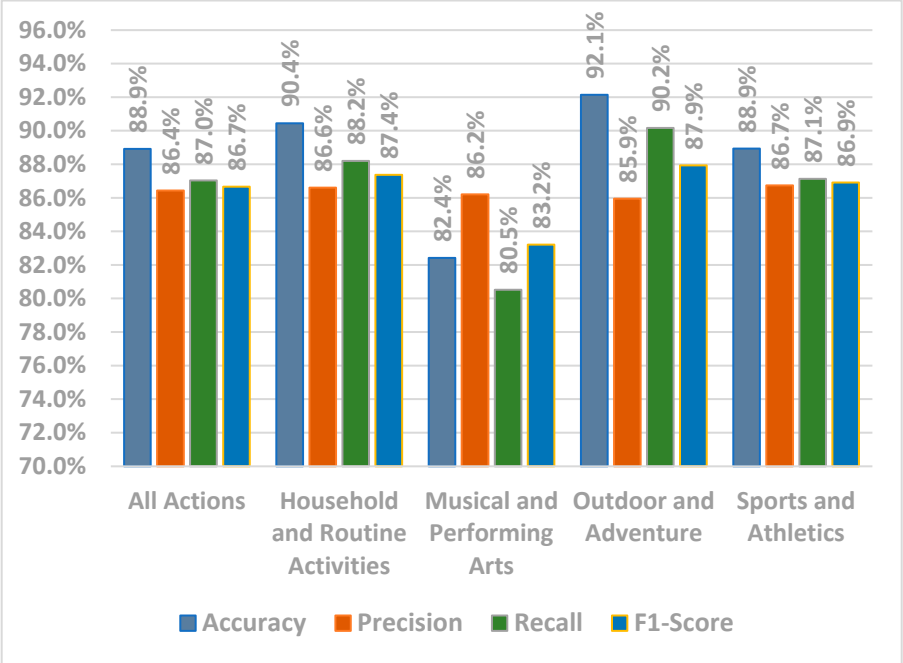

Figure 6. Overall performance of the system and by action category.

- Household and Routine Activities.

In this category, which includes actions such as mopping the floor, brushing teeth, and cutting in the kitchen, the system achieved an accuracy of 90.4%, the highest among non-dynamic action categories. The system's precision was 86.6%, and recall was 88.2%, demonstrating its capability to correctly identify most routine activities with relatively low false positives and false negatives. The F1-score of 87.4% highlights the system's balanced performance in recognizing routine actions that involve subtle joint movements. This category's strong performance indicates that the system is well-suited to handle less dynamic but repetitive activities.

- Musical and Performing Arts.

The system faced more challenges in this category, where actions such as playing the cello, playing the guitar, or performing floor gymnastics are characterized by more intricate joint movements. The system achieved an accuracy of 82.4%, the lowest among

all categories. The precision was 86.2%, but the recall was lower at 80.5%, reflecting the difficulty in detecting all instances of these actions, likely due to the interaction with tools (e.g., musical instruments) complicating joint movement recognition. The F1-score of 83.2% underscores the need for further refinement in handling complex tool-based activities.

- Outdoor and Adventure Activities.

The system performed exceptionally well in this category, achieving the highest accuracy of 92.1%. This category includes dynamic activities such as rope climbing, Tai Chi, and skiing, where distinct and large joint movements are common. The precision was 85.9%, and the recall was 90.2%, indicating the system's ability to accurately detect most of these activities with minimal false negatives. The F1-score of 87.9% further emphasizes the system's robust performance in fast-paced, high-movement actions. The clear distinction in movement patterns for outdoor activities likely contributed to the system's superior performance in this group.

- Sports and Athletics.

For actions such as basketball, archery, and weightlifting, the system achieved an accuracy of 88.9%. The precision was 86.7%, and the recall was 87.1%, demonstrating consistent performance in recognizing physically intensive sports that involve both upper and lower body coordination. The F1-score of 86.9% highlights the system's balanced handling of these activities, with only minor challenges in actions involving rapid transitions or simultaneous movements. The system's success in recognizing individual and team sports activities showcases its flexibility and adaptability to a range of athletic actions.

4.5. Comparative Performance Against Baseline and State-of-the-Art Methods

To further assess the effectiveness of the proposed HAR system, we compared its performance with a traditional baseline method, specifically, a CNN-based model, which is widely used for HAR tasks.

The baseline CNN model consists of three consecutive convolutional layers with 32, 64, and 128 filters, respectively, followed by max pooling layers and two fully connected layers. This architecture was chosen as the representative standard for HAR tasks. The baseline model demonstrates the standard approach in which video frames are processed directly without any extraction of features of joint movements via OpenPose, as proposed in the method. Given that the input was a video file that we only processed to 64×64 size, the input layer was set to 64×64. As the baseline model does not employ pose-based features for feature extraction, it operates solely on raw video data, thus allowing for an evaluation of the proposed graph-based method.

The Firefly Optimization Algorithm was applied to fine-tune the hyperparameters of both the baseline CNN model and the graph-based HAR system to ensure a fair comparison. CNN-based models are commonly adopted in HAR due to their ability to automatically extract spatial features from video streams. However, while CNNs excel at capturing spatial relationships within individual frames, they are less effective at modeling the temporal dynamics inherent in videos. The results of the comparison are illustrated in Figure 7.

In our experiments, the proposed graph-based HAR system outperformed the standard CNN model across all key performance metrics. The overall accuracy of the graph-based system was 88.9%, which exceeded the CNN-based model's accuracy of 83.7%. This improvement in accuracy can be attributed to the graph structure's ability to capture both spatial and temporal relationships between human joints more effectively than traditional models that process video frames sequentially without explicitly modeling these connections.

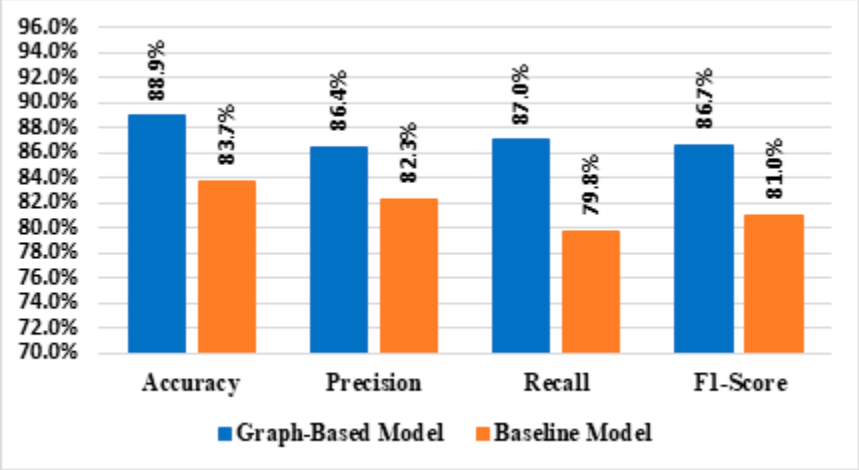

Figure 7. Results of the optimized graph-based model and baseline CNN model.

The precision and recall metrics followed similar trends. The graph-based approach achieved a precision of 86.4% and recall of 87.0%, compared to 83.0% precision and 82.5% recall for the CNN model. These results demonstrate the graph-based system's enhanced capability to reduce false positives while maintaining a higher recall rate, allowing it to better detect human activities, even in complex scenarios.

The F1-score of 86.7% achieved by the graph-based HAR system also outperformed the CNN's F1-score of 82.7%. This indicates that the proposed system offers a balanced performance between precision and recall, which is crucial for practical applications where minimizing both false positives and false negatives is important.

A key reason for the superior performance of the proposed system lies in its use of a graph structure to model human joint movements, which enables the capture of both spatial relationships (the position of joints relative to one another) and temporal relationships (the movement of joints over time). This graph-based representation allows the system to better understand and classify complex actions that involve joint transitions, interactions with objects, and multi-step movements, which are challenging for conventional models that treat video frames independently.

Moreover, to evaluate the efficacy of the proposed method, we compared it against several state-of-the-art (SOTA) approaches in human activity recognition (HAR), including CNN–Transformer hybrid models, 3D convolutional neural networks (3DCNNs), and deeply transferred Motion Vector CNNs (DTMV-CNNs) using the UCF-101 dataset. Table 5 summarizes the results across metrics such as accuracy, precision, recall, and F1-score. Our proposed method, a graph-based CNN (GB-CNN), achieved the highest accuracy of 89.6%, outperforming CNN-Transformer models (87.5%, Shi & Liu, 2024 [20]), 3DCNNs (79.9%, Vrskova et al., 2022 [18]), and DTMV-CNNs (86.4%, B. Zhang et al., 2018 [16]).

Table 5. Comparison of the proposed graph-based CNN (GB-CNN) with state-of-the-art methods on the UCF-101 dataset.

Reference	**Method**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
[20]	CNN+ Transform	87.5%			
[18]	3DCNN	79.9%			
[16]	DTMV-CNN	86.4%			
Proposed Approach	GB-CNN	89.6%	88.3%	87.9%	88.1%

Additionally, the proposed GB-CNN achieved superior performance across all evaluated metrics, including a precision of 88.3%, a recall of 87.9%, and an F1-score of 88.1%.

This highlights the effectiveness of leveraging pose-based graph structures, which offer enhanced robustness to occlusions and variability in joint movements. By focusing on both spatial and temporal relationships between joints, the proposed approach demonstrated improved generalization capabilities and computational efficiency compared to other SOTA methods. These results establish the proposed method as a reliable and efficient solution for HAR tasks.

In addition, we performed statistical analyses including a paired *t*-test to compare the accuracy, precision, recall, and F1-score achieved by the proposed GB-CNN method to the accuracy, precision, recall, and F1-score of the baseline models and the state-of-the-art approaches, which validate the significance of the improvement achieved by the proposed method. The test *p*-values indicate that all improvements are statistically significant ($p < 0.05$) across all metrics.

4.6. Computational Cost and Real-Time Feasibility

The computational complexity of the proposed system was analyzed to evaluate its scalability and feasibility. Table 6 below summarizes the time complexity of the key stages in the system, including preprocessing, graph construction, model training, and inference.

Table 6. Computational complexity of the proposed system.

Phase	**Complexity**
Preprocessing (Pose Estimation)	$O(F \cdot W \cdot H \cdot D)$
Graph Construction	$O(F \cdot N^2)$
Model Training	$O(E \cdot F \cdot N^2 \cdot D_c)$
Inference	$O(F \cdot N^2 \cdot D_c)$

Here, $\boldsymbol{F}$ represents the number of video frames, $\boldsymbol{W} \cdot \boldsymbol{H}$ denotes the frame resolution, $\boldsymbol{N}$ is the number of keypoints in the pose graph, $\boldsymbol{D}$ and $\boldsymbol{D_c}$ are the network depths for pose estimation and the graph-based CNN, respectively, and $\boldsymbol{E}$ is the number of training epochs.

Preprocessing by OpenPose is the heaviest process by far among the steps as it depends on the resolution of the input frame ($\boldsymbol{W} \cdot \boldsymbol{H}$), and model training is second, which is realized by the number of epochs ($\boldsymbol{E}$) and graph nodes ($\boldsymbol{N}$). Graph construction, while quadratic in complexity with respect to $\boldsymbol{N}$, is manageable due to the limited number of joints in typical datasets. The inference step is computationally efficient, making the system feasible for real-time applications, especially with optimized hardware. This breakdown provides a clear understanding of the system's scalability and practical implementation requirements.

Real-time implementation of the proposed system was investigated on a computer with NVIDIA RTX 3050 GPU, Intel i5 11th Generation Processor, and 16 GB RAM. The timings reported demonstrate a total of about 45 ms/frame (30 ms/frame for preprocessing–pose estimation using OpenPose; 5 ms/frame for graph construction; 10 ms/frame for inference). The processing speed of about 22 frames per second (FPS) is more than needed for real-time applications such as video surveillance and human–computer interactions.

The results demonstrate that the system is capable of operating in real-time on a moderately powerful computer, balancing computational efficiency with high accuracy. These findings confirm the system's practicality for deployment in real-world scenarios.

5. Conclusions

This research introduced a novel human activity recognition (HAR) system that integrates graph structures and deep neural networks to address the challenges inherent in activity recognition. By leveraging graph representations, we captured both spatial relationships and temporal dynamics of joint movements, resulting in a more nuanced

understanding of human activities. The application of the Firefly Optimization Algorithm further enhanced the system's performance by fine-tuning critical hyperparameters, leading to significant improvements in accuracy, precision, recall, and F1-score.

The experimental results demonstrated the effectiveness of the proposed graph-based HAR system, achieving superior performance compared to traditional CNN-based models. Our system achieved an overall accuracy of 88.9%, with robust performance across diverse action categories, including sports, routine activities, and musical performances. The optimized system not only improved training accuracy but also minimized the gap between training and validation performance, highlighting its enhanced generalization ability and reduced overfitting.

Additionally, the categorization of actions revealed that the system performed exceptionally well in dynamic activities, such as outdoor sports and adventure, while still maintaining reliable performance in more intricate activities involving tools, such as musical performances. These results emphasize the versatility and adaptability of the proposed approach for real-world applications.

The effectiveness of the proposed GB-CNN model is further confirmed by comparing it with state-of-the-art methods, which show higher accuracy and precision, recall, and F1-score on the UCF-101 dataset. The proposed method is both occlusion- and joint-movement-robust while being efficiently computable by using pose-based graphs. The results highlight the potential utility of the presented approach in many actual world applications like surveillance, healthcare, and human–computer interactions. The system's ability to process frames at 22 FPS on a moderately equipped computer confirms its feasibility for real-time applications, making it suitable for scenarios such as surveillance, healthcare, and human–computer interactions.

The findings of this research contribute to the growing body of knowledge in the field of HAR by presenting a system that offers both accuracy and robustness. Future work could explore further refinements to the graph-based model, such as incorporating multi-modal data sources or enhancing the handling of occlusions in complex scenarios. With continued development, the proposed HAR system holds the potential to impact various fields, including healthcare monitoring, smart surveillance, and sports analytics, enabling safer, more intuitive, and privacy-preserving activity recognition solutions.

Funding: This research received no external funding.

Data Availability Statement: Data sharing does not apply to this article as no datasets were generated or analyzed during the current study.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Makikawa, M.; Iizumi, H. Development of an ambulatory physical activity memory device and its application for the categorization of actions in daily life. *Medinfo. MEDINFO* **1995**, *8 Pt 1*, 747–750. [PubMed]
2. Amer, A.; Ji, Z. Human locomotion activity recognition using spectral analysis and convolutional neural networks. *Int. J. Manuf. Res.* **2021**, *16*, 350–364. [CrossRef]
3. Dharejo, F.A.; Zawish, M.; Zhou, Y.; Davy, S.; Dev, K.; Khowaja, S.A.; Fu, Y.; Qureshi, N.M.F. FuzzyAct: A Fuzzy-Based Framework for Temporal Activity Recognition in IoT Applications Using RNN and 3D-DWT. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4578–4592. [CrossRef]
4. Jana, G.C.; Swetapadma, A.; Pattnaik, P.K. A hybrid method for classification of physical action using discrete wavelet transform and artificial neural network. *Int. J. Bioinform. Res. Appl.* **2021**, *17*, 25. [CrossRef]
5. Zhuang, W.; Xu, S.; Han, Y.; Su, J.; Gao, C.; Yang, D. The design and implementation of a wearable human activity recognition system based on IMU. *Int. J. Embed. Syst.* **2020**, *13*, 158–168. [CrossRef]
6. Alazeb, A.; Azmat, U.; Al Mudawi, N.; Alshahrani, A.; Alotaibi, S.S.; Almujally, N.A.; Jalal, A. Intelligent localization and deep human activity recognition through IoT devices. *Sensors* **2023**, *23*, 7363. [CrossRef] [PubMed]

7. Xu, S.; Zhang, L.; Tang, Y.; Han, C.; Wu, H.; Song, A. Channel attention for sensor-based activity recognition: Embedding features into all frequencies in DCT domain. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12497–12512. [CrossRef]
8. Preece, S.J.; Paul, L.; Kenney, J.; Howard, D.; Goulermas, J.Y.; Kenney, L.P.J. A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. *IEEE Trans. Biomed. Eng.* **2009**, *56*. [CrossRef]
9. Aziz, A.; Mirzaliev, S.; Maqsudjon, Y. Real-time Monitoring of Activity Recognition in Smart Homes: An Intelligent IoT Framework. *J. Intell. Syst. Internet Things* **2023**, *10*, 76. [CrossRef]
10. Ketu, S.; Mishra, P.K. *Performance Analysis of Machine Learning Algorithms for IoT-Based Human Activity Recognition*; Springer: Singapore, 2020; Volume 672, pp. 579–591. [CrossRef]
11. Khaled, H.; Abu-Elnasr, O.; Elmougy, S.; Tolba, A.S. Intelligent system for human activity recognition in IoT environment. *Complex Intell. Syst.* **2023**, *9*, 3535–3546. [CrossRef] [PubMed]
12. Mishra, R. A review on learning-based algorithms for human activity recognition. *Int. J. Data Anal. Tech. Strat.* **2023**, *15*, 339–355. [CrossRef]
13. Verma, U.; Tyagi, P.; Kaur, M. Artificial intelligence in human activity recognition: A review. *Int. J. Sens. Netw.* **2023**, *41*, 1–22. [CrossRef]
14. Mohmed, G.; Lotfi, A.; Pourabdollah, A. Employing a deep convolutional neural network for human activity recognition based on binary ambient sensor data. In Proceedings of the 13th ACM international conference on pervasive technologies related to assistive environments, Corfu, Greece, 30 June–3 July 2020; ACM International Conference Proceeding Series. pp. 412–418. [CrossRef]
15. Natani, A.; Sharma, A.; Perumal, T. Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes. *Appl. Intell.* **2021**, *51*, 6014–6028. [CrossRef]
16. Zhang, B.; Wang, L.; Wang, Z.; Qiao, Y.; Wang, H. Real-Time Action Recognition With Deeply Transferred Motion Vector CNNs. *IEEE Trans. Image Process* **2018**, *27*, 2326–2339. [CrossRef]
17. Angelini, F.; Fu, Z.; Long, Y.; Shao, L.; Naqvi, S.M. 2D Pose-Based Real-Time Human Action Recognition With Occlusion-Handling. *IEEE Trans. Multimed.* **2019**, *22*, 1433–1446. [CrossRef]
18. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. Human Activity Classification Using the 3DCNN Architecture. *Appl. Sci.* **2022**, *12*, 931. [CrossRef]
19. Malik, N.U.R.; Abu-Bakar, S.A.R.; Sheikh, U.U.; Channa, A.; Popescu, N. Cascading Pose Features with CNN-LSTM for Multiview Human Action Recognition. *Signals* **2023**, *4*, 40–55. [CrossRef]
20. Shi, C.; Liu, S. Human action recognition with transformer based on convolutional features. *Intell. Decis. Technol.* **2024**, *18*, 881–896. [CrossRef]
21. Vernikos, I.; Spyropoulos, T.; Spyrou, E.; Mylonas, P. Human Activity Recognition in the Presence of Occlusion. *Sensors* **2023**, *23*, 4899. [CrossRef]
22. Narayanan, S.; Sastry, G.H.; Aswal, S.; Marriboyina, V.; Sankaranarayanan, R.; Varsha, N. Visible property enhancement techniques of IoT cameras using machine learning techniques. *Int. J. Nanotechnol.* **2023**, *20*, 569–585. [CrossRef]
23. Zhang, L.; Cui, W.; Li, B.; Chen, Z.; Wu, M.; Gee, T.S. Privacy-Preserving Cross-Environment Human Activity Recognition. *IEEE Trans. Cybern.* **2023**, *53*, 1765–1775. [CrossRef]
24. Geng, H.; Huan, Z.; Liang, J.; Hou, Z.; Lv, S.; Wang, Y. Segmentation and Recognition Model for Complex Action Sequences. *IEEE Sensors J.* **2022**, *22*, 4347–4358. [CrossRef]
25. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild. *arXiv* **2012**, arXiv:1212.0402. [CrossRef]
26. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The Kinetics Human Action Video Dataset. *arXiv* **2017**, arXiv:1705.06950. [CrossRef]
27. Yang, X.S. Firefly algorithms for multimodal optimization. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5792, pp. 169–178. [CrossRef]
28. Nabavirazavi, S.; Taheri, R.; Ghahremani, M.; Iyengar, S.S. Model Poisoning Attack Against Federated Learning with Adaptive Aggregation. *Adv. Inf. Secur.* **2024**, *104*, 1–27. [CrossRef]