technologies

MDPI

Article

Skeleton-Based Real-Time Hand Gesture Recognition Using Data Fusion and Ensemble Multi-Stream CNN Architecture

Maki K. Habib *, Oluwaleke Yusuf and Mohamed Moustafa

School of Sciences and Engineering, MENG and RCSS, The American University in Cairo (AUC), AUC Avenue, New Cairo 11835, Egypt
* Correspondence: maki@aucegypt.edu

Abstract

Hand Gesture Recognition (HGR) is a vital technology that enables intuitive human–computer interaction in various domains, including augmented reality, smart environments, and assistive systems. Achieving both high accuracy and real-time performance remains challenging due to the complexity of hand dynamics, individual morphological variations, and computational limitations. This paper presents a lightweight and efficient skeleton-based HGR framework that addresses these challenges through an optimized multi-stream Convolutional Neural Network (CNN) architecture and a trainable ensemble tuner. Dynamic 3D gestures are transformed into structured, noise-minimized 2D spatiotemporal representations via enhanced data-level fusion, supporting robust classification across diverse spatial perspectives. The ensemble tuner strengthens semantic relationships between streams and improves recognition accuracy. Unlike existing solutions that rely on high-end hardware, the proposed framework achieves real-time inference on consumer-grade devices without compromising accuracy. Experimental validation across five benchmark datasets (SHREC2017, DHG1428, FPHA, LMDHG, and CNR) confirms consistent or superior performance with reduced computational overhead. Additional validation on the SBU Kinect Interaction Dataset highlights generalization potential for broader Human Action Recognition (HAR) tasks. This advancement bridges the gap between efficiency and accuracy, supporting scalable deployment in AR/VR, mobile computing, interactive gaming, and resource-constrained environments.

Keywords: hand gesture recognition; data fusion; real-time processing; human–computer interaction; skeleton-based; multi-stream CNN; spatiotemporal representation; low-latency recognition; ensemble learning; consumer-grade hardware; computational efficiency

check for updates

Academic Editors: Fabrizio Stasolla and Everardo Inzunza-González

Received: 19 August 2025
Revised: 3 October 2025
Accepted: 6 October 2025
Published: 26 October 2025

Citation: Habib, M.K.; Yusuf, O.; Moustafa, M. Skeleton-Based Real-Time Hand Gesture Recognition Using Data Fusion and Ensemble Multi-Stream CNN Architecture. *Technologies* **2025**, *13*, 484. https://doi.org/10.3390/technologies13110484

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hand Gesture Recognition (HGR) is a pivotal technology in perceptual computing, enabling computers to interpret and respond to human hand movements through advanced mathematical models and artificial intelligence algorithms. This capability plays a critical role in applications such as human–computer interaction (HCI), behavioral analysis, assistive living technologies, augmented and virtual reality (AR/VR), and smart evironments [1]. Despite these advancements, achieving high accuracy and real-time recognition of hand gesture remains a formidable challenge due to the inherent complexity of hand gestures, variations in morphology, occlusions, background clutter, sensor noise, and computational constraints in real-world settings [2–10].

A robust HGR system must address these challenges while optimizing accuracy, computational efficiency, real-time responsiveness, and hardware adaptability. Since hand gestures are dynamic and time-dependent, models must extract spatiotemporal features from sequential hand positions to comprehensively capture the intent of the gesture. A broad spectrum of HGR frameworks has been developed using different modalities, such as RGB images, depth maps, skeleton data, and optical flow, alongside diverse architectures such as 3D Convolutional Neural Networks (3D-CNNs), Recurrent Neural Networks (RNNs), Attention Networks, and Long-Term Recurrent Convolutional Networks (LRCNs) [5,11–18]. However, existing solutions often struggle to balance recognition accuracy and real-time processing, particularly on consumer-grade hardware [19,20].

State-of-the-art (SOTA) HGR models often prioritize accuracy over computational efficiency, frequently requiring high-performance GPUs and dedicated processing units. This restricts accessibility for low-power platforms such as mobile devices, wearables, and embedded systems, where computational resources are constrained [14,15,21,22]. Furthermore, many HGR architectures rely on extensive training datasets and data augmentation strategies, limiting their adaptability for real-time, interactive applications [23,24].

Skeleton-based HGR has emerged as a promising alternative, leveraging structured, low-dimensional representations of dynamic gestures to enhance computational efficiency while preserving essential spatiotemporal features. By converting three-dimensional (3D) skeletal motion data into structured two-dimensional (2D) spatiotemporal representations, these models reframe gesture recognition as an image classification task, making them well-suited for CNN-based architectures [5,14,25,26]. Despite these advantages, existing skeleton-based HGR approaches exhibit several limitations, including noise in data fusion, suboptimal view orientations, and inadequate generalization across datasets [27–29].

One of the primary limitations of skeleton-based HGR is the reliance on data-level fusion techniques, which often yield noisy, visually indistinct RGB representations of gesture sequences, thereby increasing classification errors [25,30]. The effectiveness of selected view orientations is also critical, as suboptimal perspectives can introduce distortions in spatiotemporal representations, thereby reducing recognition accuracy. Additionally, many prior frameworks lack comprehensive benchmarking on widely recognized datasets, raising concerns about their scalability and robustness across diverse user demographics [5,14–16,31].

Moreover, many skeleton-based models fail to achieve real-time inference on standard consumer hardware due to their reliance on multi-stream processing or deep recurrent architectures, both of which impose high computational demands. These challenges limit their practical deployment in real-world applications, especially in scenarios where low-latency, resource-efficient processing is crucial [20,28,32].

To overcome these challenges, this paper presents a novel, lightweight skeleton-based HGR framework that integrates optimized data-level fusion with a multi-stream CNN architecture. The main contributions of this work include:

1. Enhanced Data-Level Fusion: A refined transformation process for converting 3D skeleton data into high-quality 2D spatiotemporal representations, reducing noise and improving gesture classification accuracy.
2. Optimized Multi-Stream CNN Architecture with a fully trainable ensemble tuner mechanism that enhances semantic connections between multiple gesture representations, leading to improved classification performance.
3. The presented framework significantly reduces computational complexity while realizing real-time performance on standard devices, maintaining the accuracy necessary and comparable to state-of-the-art methods.

4. Extensive Benchmarking and Validation: Empirical evaluations conducted on five widely used benchmark datasets (SHREC2017, DHG1428, FPHA, LMDHG, and CNR) validate the framework's robustness and generalizability across diverse gesture categories.

The conducted experiments demonstrate that the developed framework achieves competitive recognition accuracy, matching or surpassing SOTA benchmarks while operating with significantly lower computational overhead. Additionally, the SBU Kinect Interaction Dataset (SBUKID) was utilized in an exploratory study to demonstrate the framework's adaptability for broader Human Action Recognition (HAR) applications, thereby reinforcing its scalability beyond gesture recognition. This research paves the way for real-time HGR applications in mobile computing, interactive gaming, and resource-limited environments by addressing the trade-off between accuracy and efficiency.

2. Related Work

This section provides an in-depth review of key advancements in Hand Gesture Recognition (HGR), which serve as the foundation for our proposed framework. The related works are categorized into three primary approaches: skeleton-based HGR, data-level fusion of temporal information, and multi-stream network architectures.

2.1. Skeleton-Based Hand Gesture Recognition

Traditionally, HGR frameworks have predominantly relied on RGB videos or depth maps for gesture recognition. However, with the evolution of computational methodologies and an improved understanding of gesture intricacies, modern frameworks have increasingly shifted towards skeleton-based representations. Skeleton-based models, derived from RGB-D data, offer a structured approach to encoding hand positions, mitigating challenges related to occlusions, background variability, and morphological differences across individuals. Despite their advantages, these models often require offline pre-processing to extract skeletal information from RGB-D sources, which introduces susceptibility to errors during inference [15,32,33]. These limitations have also been emphasized in studies such as [34,35], which highlight the role of preprocessing variability in model robustness.

Several pivotal contributions have shaped the development of skeleton-based HGR. Temporal Convolutional Networks (TCNs) have been explored for modeling skeleton pose sequences, incorporating motion summarization modules to enhance feature learning [32]. Attention-based spatial-temporal networks have also been introduced to capture inter-joint dependencies without explicit knowledge of joint connections [15]. Furthermore, streamlined architectures incorporating Joint Collection Distances and multi-scale motion features have been proposed to enhance recognition robustness [8,36]. Deep Convolutional LSTM (ConvLSTM) models have demonstrated the ability to automatically extract discriminative spatiotemporal features by capturing gestures across multiple scales [37], as reinforced by newer models evaluated in [35].

Unimodal skeleton-based frameworks generally outperform their multimodal counterparts in terms of computational efficiency, as they reduce the number of trainable parameters and floating-point operations per second (FLOPS) [5,15,32]. These models exhibit lower architectural complexity while maintaining high recognition accuracy, making them more feasible for real-time deployment [5,8,15,21,38,39]. Our proposed approach capitalizes on the efficiency of skeleton representations while mitigating the high computational demands typically associated with deep learning-based HGR. Furthermore, skeleton-based recognition inherently minimizes privacy concerns linked to traditional RGB-based vision systems, as skeletal data does not contain identifiable facial features [1,40]. These privacy advantages are especially significant in sensitive environments such as healthcare, as explored in [35].

2.2. Data-Level Fusion

The recognition of dynamic hand gestures is inherently complex due to their temporal nature. Gestures comprise a sequence of hand movements, requiring a robust HGR framework capable of discerning temporal relationships for accurate classification. This challenge is particularly pronounced in multi-stream and multimodal networks, where different representations of the same gesture must be effectively synchronized [41]. Prior studies such as [30,35] have shown that the choice of fusion level directly impacts computational load and robustness in dynamic gesture environments.

The effectiveness of an HGR model is primarily determined by how input modalities and network streams are integrated. Deep learning models typically employ integration at three levels: data-level, feature-level, and decision-level, each offering unique trade-offs [41]. Among these, data-level fusion has gained significant attention as it allows for early integration of temporal information before deep feature extraction. This approach has been particularly useful in static image-based gesture recognition, as it enables the transformation of dynamic gesture sequences into condensed representations suitable for CNN-based classification [25,42]. In particular, [26] demonstrated how skeleton-derived temporal images can preserve motion trajectories with minimal loss, improving early-stage classification.

One widely adopted technique is Temporal Information Condensation, which transforms gesture sequences into static images before training. This method enhances the network's ability to encode spatiotemporal patterns efficiently, facilitating transfer learning using pre-trained CNN models. Optical flow-based motion fusion, as introduced in [41], supplements static RGB images with motion-derived frames, thereby improving the model's sensitivity to gesture dynamics. Another notable method, star RGB, aggregates frame-wise intensity differences to generate a single gesture-representative image, which is then classified using an ensemble of ResNet CNNs [24,31,43].

A particularly effective approach is skeleton-based data transformation, where temporal variations in skeletal joint positions are projected onto 2D spatial representations. These projections are subsequently processed using a CNN-based pipeline, as demonstrated in [25]. Although this approach has achieved promising results, further optimization in data-level fusion techniques could enhance model adaptability across more diverse datasets. Our framework incorporates an improved version of temporal condensation to minimize information loss while maintaining computational efficiency. This direction aligns with insights from [26,30], which advocate for hybrid condensation techniques that preserve both temporal dynamics and geometric coherence in a lightweight format.

2.3. Multi-Stream Network Architectures

Advanced HGR frameworks often employ multi-stream network architectures to enhance recognition accuracy by integrating multiple spatiotemporal representations. These architectures operate across different temporal scales, process diverse input modalities, and provide comprehensive gesture embeddings during training [2,15,22,30,32,44,45]. Multi-stream networks effectively capture gesture context from multiple perspectives, improving class separability and reducing the risk of overfitting. For example, studies such as [46,47] demonstrated the use of LRCNs and multi-headed attention blocks to achieve synchronized learning across temporal domains.

Multi-stream models utilize two primary fusion mechanisms: feature-level fusion, where extracted feature maps from different convolutional layers are combined, and decision-level fusion, which merges classification probabilities from independent streams. These techniques ensure the balanced integration of information from each sub-network, thereby enhancing the overall model's robustness. Recent attention-aware feature fu-

sion techniques have also been proposed to improve cross-modal understanding, such as in [1], where alignment between RGB and skeleton data significantly boosted gesture disambiguation.

Among the pioneering multi-stream models, layered 3D-CNN architectures have demonstrated significant potential. For instance, ref. [2] utilized a hierarchical structure where the first 3D-CNN detects coarse gestures, triggering a secondary CNN for fine-grained classification. Similarly, multimodal frameworks like [44] utilize independent I3D sub-networks to handle different modalities, enforcing feature consistency through semantic alignment constraints. Another study [22] introduced a dual-channel CNN architecture that processes high-resolution and low-resolution gesture sequences in parallel, with residual branches enhancing feature propagation.

While multi-stream architectures offer significant advantages, they are often computationally intensive. Lightweight 3D CNN variants, such as Inception-ResNet with separable convolutions, have been proposed for real-time gesture recognition [48]. Recent studies have also introduced efficient alternatives that integrate data-level fusion with single-stream CNNs, achieving comparable accuracy at lower computational costs [25]. Additionally, domain-aware and attention-based mechanisms, such as the Spatio-Temporal Domain-Aware Network (STDA-Net), have been explored to enhance spatiotemporal feature learning and improve classification performance in skeleton-based action recognition tasks [31]. Our proposed framework builds upon these advancements by refining the multi-stream pipeline and optimizing the fusion mechanisms to enhance recognition efficiency and minimize processing overhead.

3. Skeleton-Based Hand Gesture Recognition Framework

This section elaborates on the core components of the proposed skeleton-based hand gesture recognition framework, as depicted in Figure 1. It begins by examining the method of condensing temporal information to generate static, spatiotemporal RGB images. Next, it introduces the network architecture that utilizes this data-level fusion strategy. Finally, it explains how the combination of data-level fusion and network architecture forms a hand gesture recognition framework that is adaptable to any source of hand skeleton data, whether from real-time inference or compiled datasets.

3.1. Generation of Static Spatiotemporal Images

This level of data fusion represents the primary processing phase, performed either offline or in real time, where information from both the spatial and temporal channels of a dynamic gesture sequence is integrated into a single static 2D spatiotemporal image with a preserved square aspect ratio. This process condenses the temporal characteristics of the sequence, while ensuring that essential semantic information about the gesture remains intact. Following a similar methodology to [10], the task of converting a dynamic gesture sequence into a static spatiotemporal image can be formally described as follows:

Let g_i represent a dynamic gesture, and let $S = G_{h}{}_{h=1}^{N}$ denote the set of gesture sequences with τ spanning N classes. The temporal evolution of g_i is expressed in Equation (1):

$$G_{i=}\{G_i^{\tau}\}_{\tau=1}^{T_i} \tag{1}$$

Within a temporal window of length $T_{i,}$, $\tau \in [1, T_i]$ refers to a specific time instant τ, and G_i^{τ} designates the frame of g_i at that moment. The task of dynamic hand gesture classification is to assign the corresponding class *Ch* to each sequence *gi* accurately.

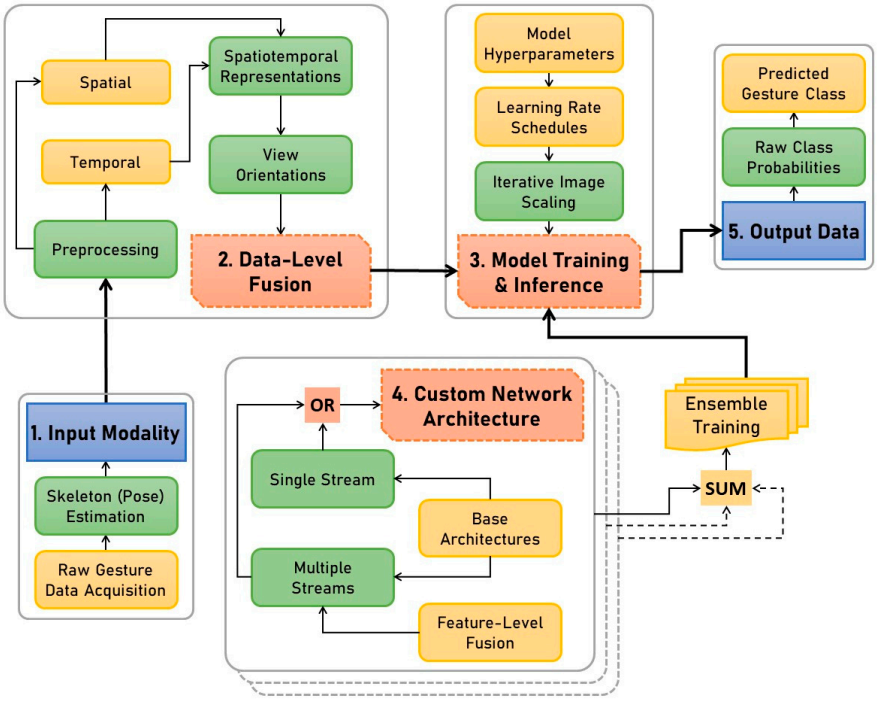


Figure 1. A schematic representation of the proposed HGR framework, illustrating the constituent modules responsible for the recognition and classification of dynamic hand gestures [10].

Gesture sequences within the set S exhibit variable temporal lengths of time windows influenced by both the gesture type and the performer's speed. To standardize input, each sequence g_i in S is resampled to a common window T_i, chosen to be longer than any individual window in the set of T_i in S. This standardization process has the following implications for each gesture sequence g_i:

- This process smooths out inaccuracies present in individual frames G_i^τ that may occur during pose estimation.
- It also reduces minor variations in motion trajectories and sequence durations caused by individual differences in gesture performance.

Together, these resampling steps emphasize the shared features among gesture sequences $g_i \in C_h$ within S, which enhances the condensation of temporal information and, as a result, increases the model's classification accuracy. After resampling, each gesture g_i is separated into spatial and temporal components:

- Spatial: This channel represents changes in hand pose across frames G_i^τ as a 3D model of the hand skeleton. Each finger is displayed in a unique CSS color, chosen for high contrast with the background and with the others.
- Temporal: This channel visualizes hand motion through 3D "temporal trails" drawn by the five fingertips, extending from the gesture's start G_i^1 to its end G_i^T. These trails consist of sequential markers, each assigned a distinctive CSS color for its respective finger. The alpha channel adjusts the transparency of each marker: those earlier in the sequence ($\tau \approx 1$) are more transparent, while those later ($\tau \approx$ T) are more opaque, thus capturing the gesture's temporal progression.

Condensing the temporal information in this way produces, for each resampled gesture g_i, a 3D spatiotemporal image. This image combines the hand's pose at the final frame G_i^T with its temporal path $G_{i\,\tau=1}^{\tau\;T-1}$, as illustrated in Figure 2. The resulting 3D spatiotemporal image can be viewed from any chosen perspective; in this work, six specific view orientations are considered: axonometric, front-away, side-left, top-down, front-to, and side-right.

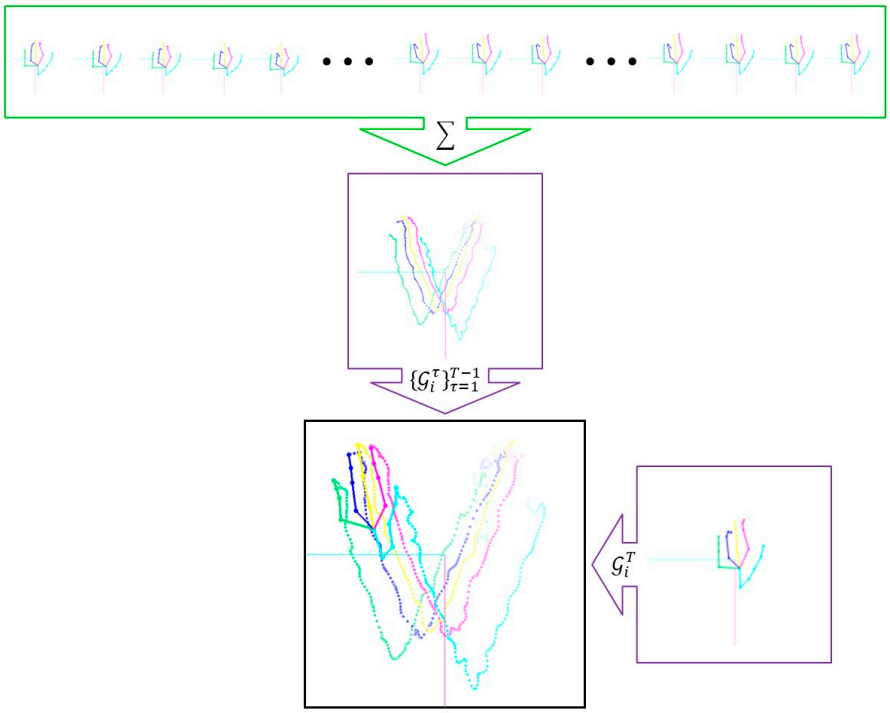


Figure 2. The data-level fusion workflow. This figure shows how 3D skeleton data for a DHG1428 Swipe-V gesture (front-to-view orientation) is transformed into its 2D spatiotemporal RGB representation for subsequent image classification.

Each VO is determined by specific elevation and azimuth angles assigned to the virtual camera during visualization, as depicted in Figure 3. The selection of these angles varies depending on the dataset and is influenced by the approach taken to capture and process skeleton data from participant gestures.

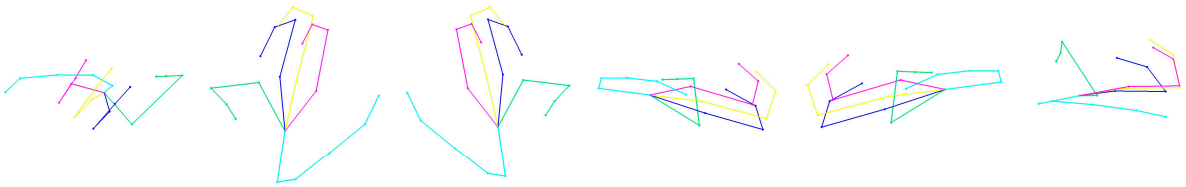


Figure 3. illustrates the six view orientations (VOs) used to visualize spatiotemporal gesture representations, shown from left to right as top-down, front-to, front-away, side-right, side-left, and axonometric.

For each gesture $g_{i,}$ the result of data-level fusion is a single image depicting its 3D spatiotemporal representation from one of the six selected view orientations VO_j, where $j = 1, \ldots, 6$. It is essential to adjust the virtual camera's zoom and position during rendering to ensure the entire spatiotemporal representation fits neatly within the static image frame, avoiding any cropping or truncation. Since the optimal settings differ for each gesture and the sequence set of S, these parameters are computed dynamically rather than assigned fixed values.

$$g_i = g_i - mean(g_i) - (L/2) \tag{2}$$

$$P = (L/2) \tag{3}$$

$$Z_i = \max(g_i) - \min(g_i) + \gamma \tag{4}$$

Equation (2) shifts each gesture's data g_i so it is centred in the image by subtracting its mean and half the image length L to fit it within the static image. This adjustment ensures the positioning of the virtual camera at P in 3D space, coincides with the static image centre for all $g_i \in S$.

Equation (3), the camera's position P is fixed at a length $L/2$ from the midpoint of the static image's dimension.

Equation (4) determines the virtual camera's zoom level Z_i of the virtual camera for each g_i based on the spatial extent of the gesture, plus an optional padding parameter γ for tweaking the estimated zoom level for all $g_i \in S$.

These calculations guarantee that the 3D spatiotemporal gesture representation is optimally centred and fully visible from the chosen viewpoint in the resulting static image.

In summary, the data-level fusion process converts the challenge of dynamic hand gesture classification into a static image classification task. By applying a mapping function Φ, the 3D skeleton data of a dynamic gesture g_i observed from any view orientation VO_j is projected into a single 2D spatiotemporal image $I_{ij} = \Phi(G_i)$. This resulting image then serves as the input to the classification model, to assign it to the correct class C_h. This strategy streamlines the gesture recognition process by making use of proven methods from image classification, thereby allowing established research and algorithms in this area to be directly applied.

3.2. Multi-View Ensemble-Tuned CNN for Spatiotemporal Gesture Image Classification

With the set L of dynamic hand gestures g_i, transformed into static spatiotemporal images I_{ij}, standard convolutional neural network (CNN) architectures become applicable for gesture classification. The proposed HGR framework also leverages transfer learning, employing deep learning CNN models originally designed for image classification tasks. Transfer learning accelerates convergence during training and streamlines prototype development within the framework. To select the most effective network backbone, we assessed various CNN models pre-trained on the ImageNet dataset, each with established state-of-the-art (SOTA) performance.

In the transfer learning process, the fully connected (FC) layers of the pre-trained model are adapted to suit the new classification task. Specifically, we replaced the original FC layers with a custom classifier, while retaining the core convolutional layers (and their pre-trained weights) to act as an encoder. As illustrated in Figure 4, the new classifier comprises additional pooling, batch normalization, dropout, linear, and activation layers, all of which are trained from scratch. This strategy enables the encoder's feature maps to be more effectively repurposed for the gesture recognition task. The classifier then outputs a set of probabilities representing the likelihood that the input image I_{ij} belongs to any class $C_h,\ h \in [1,\ N]$ within the set S.

To fully utilize the multiple view orientations generated during data-level fusion, the framework employs a specialized multi-stream CNN architecture, depicted in Figure 5. This architecture integrates transfer learning with ensemble training, enabling the classification of spatiotemporal images derived from any view orientation VO_j. For each gesture $g_i \in S$, data-level fusion yields a collection of spatiotemporal images corresponding to j different view orientations.

Each image I_{ij} is sequentially passed into the multi-stream encoder, which extracts feature representations from the data. These feature maps are subsequently processed by the multi-stream classifier, as depicted in Figure 4 (right), resulting in a set of class probabilities $\{CP\}^{h\ N}_{ij\,(h=1)}$. By sharing the encoder and classifier components across all input images, the network significantly reduces its overall computational requirements. However, this approach has a limitation: the sequence in which the images (corresponding to different

view orientations) are fed into the network is fixed rather than randomized. As a result, the search space for determining the optimal order of view orientations becomes substantially larger. Within this framework, the multi-stream encoder and classifier sub-networks work together to learn the optimal combination of weights for fusing information from each view orientation VOj, with the learning process guided by the loss computed for each image I_{ij}. Because each image contributes a distinct loss value, the specific order and selection of input spatiotemporal images directly influence the overall performance of the multi-stream sub-network [10].

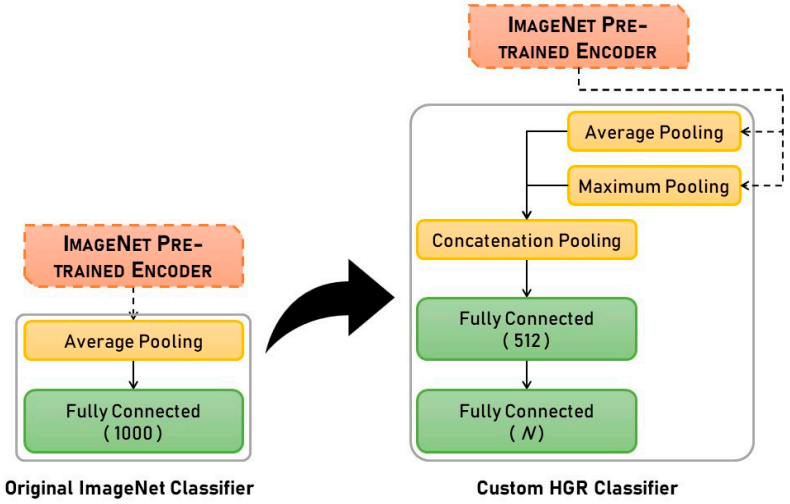


Figure 4. Comparison between the standard ImageNet classifier (**left**) and the customized classifier designed for HGR tasks (**right**).

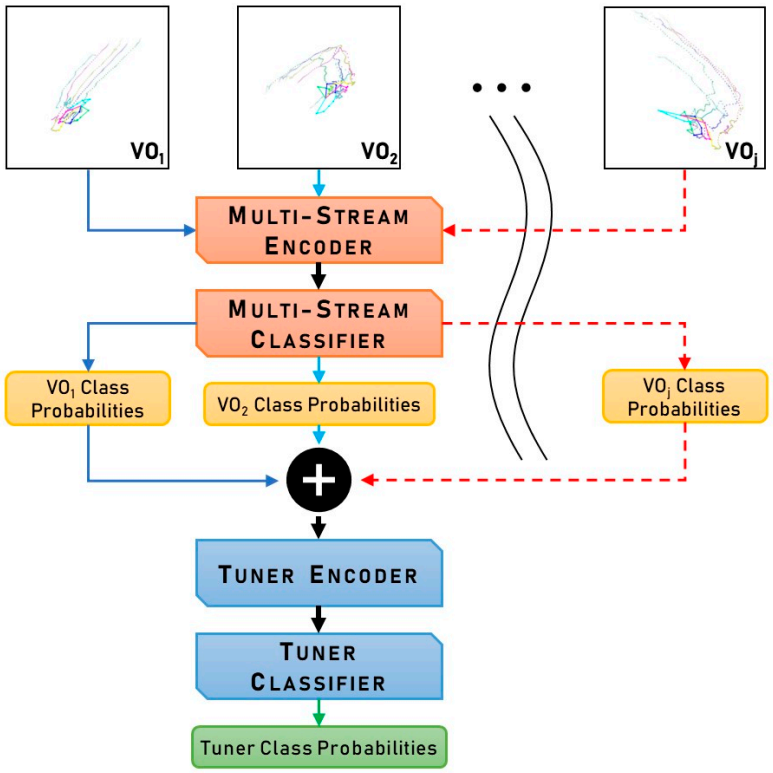


Figure 5. Overview of the Ensemble Tuner Multi-Stream CNN Framework. This architecture demonstrates how spatiotemporal images generated through data-level fusion are processed by multiple CNN backbones for hand gesture recognition.

The set of class probabilities $\{CP\}_{ij_{(h=1)}}^{h\,N}$ produced by the multi-stream classifier is then combined into a single RGB pseudo-image through an online decision-level fusion process. This pseudo-image is subsequently passed to the ensemble tuner sub-network, which generates an additional set of class probabilities. The ensemble tuner employs a streamlined, pre-trained CNN backbone as its encoder, while retaining the same classifier structure depicted in Figure 4 (right). The entire ensemble tuner multi-stream CNN architecture is trained end-to-end, resulting in $(j+1)$ class probabilities and associated loss values for each gesture g_i. In this work, we report only the class probabilities and classification accuracies yielded by the ensemble tuner sub-network. To address the multi-task aspect of the framework, we utilize a specialized loss function inspired by [49]. Mirroring the strategy in [10], this loss formulation incorporates the homoscedastic uncertainties linked to the view orientations and the ensemble tuner, assigning appropriate weights to the $(j+1)$ cross-entropy losses.

Adopting a multi-stream network that processes multiple view orientations enables the model to differentiate between gesture classes that may appear similar from a single viewpoint, thereby enhancing classification accuracy. Furthermore, using RGB pseudo-images for decision-level fusion supports transfer learning. It helps maintain semantic consistency among the class probabilities $\{CP\}_j$ generated for each view orientation $\{VO\}_j$ by the multi-stream sub-network. By integrating the multi-stream and ensemble tuner sub-networks within a unified architecture, we eliminate the need to train several models independently, a common drawback of traditional ensemble learning methods.

4. Experiments & Results

4.1. Overview of Datasets

In this section, we present the five benchmark datasets employed for developing and evaluating our proposed HGR framework, as well as for comparative analysis against current state-of-the-art (SOTA) methods. Each dataset introduces unique challenges and opportunities for advancing hand gesture recognition research.

- **CNR:** The CNR dataset [25] comprises spatiotemporal images captured from a top-down perspective, totaling 1925 gesture sequences across 16 gesture classes. The data is partitioned into a training set of 1348 images and a validation set of 577 images. Notably, this dataset does not include raw skeleton data, limiting our framework's training to a single viewpoint.
- **LMDHG:** The LMDHG dataset [50] consists of 608 gesture sequences spanning 13 classes, divided into 414 sequences for training and 194 for validation. It features minimal overlap in subjects between subsets and provides comprehensive 3D skeleton data with 46 hand joints per hand.
- **FPHA:** Containing 1175 gesture sequences distributed across 45 classes, the FPHA dataset [51] offers a diverse range of styles, perspectives, and scenarios. The main challenges include highly similar motion patterns, varied object interactions, and a relatively low ratio of gestures to classes. It is split into 600 gestures for training and 575 for validation, with 3D skeleton data available for 21 joints per subject.
- **SHREC2017:** This dataset [8] features 2800 gesture sequences performed by 28 subjects, designed for both coarse- and fine-grained classification via 14-gesture (14 G) and 28-gesture (28 G) benchmarks. The data provides 3D skeleton information for 22 hand joints and follows a 70:30 random-split protocol for training (1960 gestures) and validation (840 gestures).

- **DHG1428:** Structured similarly to SHREC2017, the DHG1428 dataset [52] comprises 2800 sequences executed by 20 subjects for both the 14 G and 28 G tasks. It provides equivalent skeleton data and employs a 70:30 split for training and validation sets.
- **SBUKID:** The SBUKID [53], a smaller human action recognition (HAR) collection, includes 282 action sequences across eight classes involving two-person interactions. It provides skeleton data for 15 joints per subject and utilizes a five-fold cross-validation protocol, reporting average accuracies across all folds.

To support a structured and transparent comparison of the six benchmark datasets used in this study, we provide a consolidated summary in Table 1. The table outlines the critical attributes for each dataset, including the total number of gesture sequences, training-validation split, number of gesture classes, number of subjects, availability of skeleton data, and validation protocol. This comparison enhances the interpretability of the evaluation outcomes and contextualizes dataset-specific constraints that influence the generalizability and robustness of the proposed HGR framework.

Table 1. Summary of Benchmark Datasets Used in This Study.

Dataset	Sequences (Train/Val)	Gesture Classes	Subjects	Skeleton Data	Validation Protocol
CNR	1925 (1348/577)	16	10	Not available (depth-based, continuous gesture)	Fixed split (top-view)
LMDHG	608 (414/194)	13	14	46 joints	Subject-wise split
FPHA	1175 (600/575)	45	6	21 joints	Fixed split (per activity)
SHREC2017	2800 (1960/840)	14 G/28 G	28	22 joints	Random 70:30 split
DHG1428	2800 (1960/840)	14 G/28 G	20	22 joints	Random 70:30 split
SBUKID	282 (Cross-val)	8	7 pairs	15 joints	5-fold cross-validation

Although modern object detection frameworks, such as YOLO, have demonstrated impressive real-time performance, their design is primarily suited for bounding-box detection rather than for classifying preprocessed spatiotemporal images. Our use of a multi-stream CNN maintains high classification accuracy while minimizing overhead for low-resource systems.

4.2. Generalized HGR Framework Evaluation

Our generalized HGR framework integrates data-level fusion to generate static 2D spatiotemporal images, combined with an ensemble tuner multi-stream CNN architecture for classification. As summarized in Table 2, the elevation and azimuth angles for the virtual camera during data-level fusion are tailored for each dataset. A padding value of $\gamma = 0.125$ is applied during sequence fitting for all datasets, except CNR, which only provides static images.

Table 2. Virtual Camera (elevation, azimuth) Angles (in degrees) For Each View Orientation.

View Orientation	DHG1428	SHREC2017	FHPA	LMDHG
top-down	(0.0, 0.0)	(0.0, 0.0)	(90.0, 0.0)	(0.0, 0.0)
front-to	(90.0, 180.0)	(90.0, 180.0)	(0.0, 180.0)	(−90.0, −180.0)
front-away	(−90.0, 0.0)	(−90.0, 0.0)	(0.0, 0.0)	(90.0, 0.0)
side-right	(0.0, −90.0)	(0.0, −90.0)	(0.0, 90.0)	(0.0, 90.0)
side-left	(0.0, 90.0)	(0.0, 90.0)	(0.0, −90.0)	(0.0, −90.0)
custom	(30.0, −132.5)	(30.0, −132.5)	(25.0, 115.0)	(−15.0, −135.0)

To select an optimal network, we evaluated 26 CNN architectures from the ResNet, Inception, EfficientNet, ResNeXt, SE-ResNeXt, SE-ResNet, and xResNet families (see Table 3). All architectures were pre-trained on ImageNet and further trained using static spatiotemporal images generated from the DHG1428 dataset, explicitly using the front-to-view orientation. Based on empirical results from two training setups (TS1 and TS2), ResNet-50 and ResNet-18 were chosen as the base encoders for the multi-stream and ensemble tuner sub-networks.

Table 3. Comparative Analysis of Various Pre-trained CNN Architectures.

CNN Architecture		**Classification Accuracies (%)**			
Family	**Variants**	**TS1**	**TS2**	**Variant Average**	**Family Average**
ResNet	ResNet18	0.8083	0.7762	0.79225	0.8149
	ResNet34	0.8190	0.7952	0.8071	
	ResNet50	0.8417	0.8012	0.82145	
	ResNet101	0.8226	0.8286	0.8256	
	ResNet152	0.8310	0.8250	0.828	
Inception	Inception-v3	0.8155	0.7762	0.79585	0.8124
	Inception-v4	0.8310	0.7964	0.8137	
	Inception-ResNet-v1	0.8179	0.8262	0.82205	
	Inception-ResNet-v2	0.8167	0.8190	0.81785	
EfficientNet	EfficientNet-B0	0.8143	0.7964	0.80535	0.8083
	EfficientNet-B3	0.8107	0.8107	0.8107	
	EfficientNet-B5	0.8143	0.8131	0.8137	
	EfficientNet-B7	0.7893	0.8179	0.8036	
ResNeXt	ResNeXt26	0.8048	0.7726	0.7887	0.8042
	ResNeXt50	0.8298	0.7952	0.8125	
	ResNeXt101	0.8226	0.8000	0.8113	
SE-ResNeXt	SE-ResNeXt50	0.8143	0.7595	0.7869	0.8033
	SE-ResNeXt101	0.8405	0.7988	0.81965	
SE-ResNet	SE-ResNet18	0.7881	0.7774	0.78275	0.8032
	SE-ResNet26	0.8048	0.7726	0.7887	
	SE-ResNet50	0.8179	0.8024	0.81015	
	SE-ResNet101	0.8310	0.8131	0.82205	
	SE-ResNet152	0.8060	0.8190	0.8125	
xResNet	xResNet50	0.7440	0.7333	0.73865	0.7379
	xResNet50-Deep	0.7226	0.7357	0.72915	
	xResNet50-Deeper	0.7405	0.7512	0.74585	

The ideal sequence of view orientations (VOs) for feeding spatiotemporal inputs into the multi-stream sub-network varies across datasets. The nature of gesture data collection and processing influences it. Our experiments determined that combinations of three unique VOs generally provide robust classification performance. The optimal sequence was identified through an iterative process: first, each VO was evaluated in a single-stream network; then, the best-performing pairs were evaluated in two-stream networks; and finally, the top triples were evaluated in a three-stream configuration. This method allowed us to tailor the VO sequence to each dataset.

A supplementary video demonstrating the gesture-to-image transformation and the real-time application is publicly available at:

https://github.com/Outsiders17711/e2eET-Skeleton-Based-HGR-Using-Data-Level-Fusion?tab=readme-ov-file (14 October 2025).

The software generated with pipreqs

opencv_python == 4.5.3.56

matplotlib == 3.3.3

pytorchcv == 0.0.67
geffnet == 1.0.2
vispy == 0.9.3
scipy == 1.5.4
torch == 1.8.1 + cu102
fastai == 2.5.3
numpy == 1.20.2
playsound == 1.2.2
mediapipe == 0.8.7.1
ipython == 8.10.0
secrets == 1.0.2
tensorboard == 2.8.0

4.3. Implementation and Training Details

This section outlines the full implementation pipeline, including the hardware-software stack, model training workflow, and reproducibility safeguards. It also highlights key architectural parameters, image processing routines, and computational benchmarks achieved during inference. Together, these details ensure technical transparency and support the replicability of our results across environments and use cases.

4.3.1. Environment and Software Setup

To ensure consistent model development and training, the framework was deployed on two environments: a local Windows 10 workstation with an NVIDIA GeForce GTX 1650 GPU (4 GB VRAM) and a remote Ubuntu 18.04.5 server with four NVIDIA GeForce GTX TITAN X GPUs (12 GB VRAM each). Synchronization between local and remote environments was maintained using Anaconda3 and Visual Studio Code.

The entire framework was implemented in Python 3.8.5. Major libraries used include:

a. PyTorch 1.8.1 + cu102 and FastAI 2.5.3 for model design, training, and evaluation.
b. OpenCV 4.0.1 and Vispy 0.9.2 for generating pseudo-images and visualizations during data-level fusion.
c. TensorBoard 2.8.0 was integrated into FastAI for visualizing metrics and logging experiments.

Pre-trained CNNs were imported from:

a. torchvision.models
b. FastAI == 2.5.3

fastai.vision.models.all

a. pytorchcv.model_provider
b. geffnet (GenEfficientNet 1.0.2)

4.3.2. Model Training and Hyperparameters

Training was carried out in staged cycles with progressively increasing input sizes (224 px, 276 px, 328 px, 380 px). The Adam optimizer was used alongside a custom composite loss function combining cross-entropy and homoscedastic uncertainty. Learning rates were initialized using FastAI's learner.lr_find() and scheduled via cosine annealing.

Each training session used a batch size of 16. Data augmentation included random horizontal flips, affine and perspective transforms, zooms, rotations, and brightness/contrast/color jitter to support generalization. Gesture sequences were temporally resampled to a fixed window of T = 250 frames, and static and pseudo-images were rendered at 960 × 960 and 224 × 224 pixels, respectively.

4.3.3. Reproducibility and Evaluation Protocol

To ensure reproducibility, random seeds were fixed across all frameworks (PyTorch, NumPy, Python random module, and CUDA). This step controlled for non-determinism arising from data shuffling, network initialization, and augmentation.

Subject-independent evaluation protocols were used for all datasets. No individual appeared in both the training and testing sets, ensuring robust generalization. Validation subsets were drawn from training data using block-wise sampling to avoid sequence fragmentation. Early stopping was monitored based on validation accuracy.

4.3.4. Inference Speed and Performance

During deployment testing, the average inference speed for the final trained model was 32.7 FPS on a single TITAN X GPU and 17.4 FPS on the GTX 1650 (local machine), for full gesture sequence processing. Model loading and pre-processing time added 0.2–0.3 s per sample, well within interactive response thresholds for real-time Human Gesture Recognition applications.

4.3.5. Reproducibility and Evaluation Protocol

This subsection outlines the subject-independent evaluation strategy, validation sampling, and measures taken to ensure reproducibility. These steps support fairness, generalizability, and experimental transparency.

To prevent data leakage and evaluate generalizability across unseen individuals, all experiments were conducted using a subject-independent evaluation protocol. For each dataset, participants were assigned exclusively to either the training or testing set, ensuring that no subject appeared in both splits. This approach follows best practices in gesture recognition literature [18,53].

Validation sets were selected from the training set using temporal block-wise sampling to preserve sequence continuity and avoid fragment-level overfitting. Early stopping was applied based on validation accuracy. For reproducibility, random seeds were fixed across all experiments.

4.4. Results on the CNR Dataset

Table 4 presents a comparative evaluation of our framework with other HGR frameworks on the CNR dataset. Our approach achieved slightly lower accuracy than the SOTA [24], with a difference of −1.73%. This decrease is attributed to the absence of raw skeleton data, which limits the ability to leverage advanced data-level fusion strategies, including denoising, sequence fitting, and multi-view orientation augmentation.

Table 4. Comparison of Validation Accuracy with SOTA on the CNR Dataset.

Method	Classification Accuracy (%)
Proposed Framework	97.05
Lupinetti et al. [54]	98.78

4.5. Results on the LMDHG Dataset

Table 5 presents the comparative results between our framework and other leading methods on the LMDHG dataset. Here, our framework outperformed the SOTA by +6.86%, a gain resulting from our improved data-level fusion approach for spatiotemporal image generation. These results underscore the effectiveness of our enhancements and demonstrate the utility of transforming dynamic gesture recognition into a static image classification problem. Notably, optimal use of custom, front-away, and top-down view orientations contributed to this improved performance.

Table 5. Comparison of Validation Accuracy with SOTA on the LMDHG Dataset.

Method	Classification Accuracy (%)
Boulahia et al. [50]	84.78
Lupinetti et al. [25]	92.11
Mohammed et al. [37]	93.81
Proposed Framework	98.97

4.6. Results on the FPHA Dataset

The FPHA dataset presents significant challenges due to its diversity of styles and strict 1:1 train/validation protocol. Table 6 shows that, despite employing the best sequence of view orientations ([front-away, custom, top-down]), our framework fell short of the SOTA [32] by −4.10%. Lupinetti et al. [54] also reported strong performance on this dataset, achieving 98.33% accuracy using CNNs with Leap Motion data. The complexity of gesture variation and similarity in motion patterns present substantial hurdles in this dataset. Alternatively, appearance-based models such as PAN [55] emphasize efficient recognition via persistence of appearance, though they are less tailored to depth or skeleton-based gesture datasets like FPHA.

Table 6. Comparison of Validation Accuracy with SOTA on the FPHA Dataset.

Method	Classification Accuracy (%)
Sahbi [29]	86.78
Liu et al. [14]	89.04
Li et al. [5]	90.26
Liu et al. [6]	90.96
Proposed Framework	91.83
Nguyen et al. [26]	93.22
Rehan et al. [30]	93.91
Sabater et al. [32]	95.93

4.7. Results on the SHREC2017 Dataset

As shown in Table 7, our framework achieved validation accuracies of 97.86% (14 G) and 95.36% (28 G) on SHREC2017, with respective improvements of +0.86% and +1.46% over SOTA [15]. These results were obtained using the optimal sequence of [front-away, custom, front-to] view orientations, further supporting the benefit of our tailored VO selection.

Table 7. Comparison of Validation Accuracy with SOTA on the SHREC2017 Dataset.

Method	Classification Accuracy (%)		
	14 G	28 G	Average
Sabater et al. [32]	93.57	91.43	92.50
Chen et al. [34]	94.40	90.70	92.55
Yang et al. [8]	94.60	91.90	93.25
Liu et al. [6]	94.88	92.26	93.57
Liu et al. [14]	95.00	92.86	93.93
Rehan et al. [30]	95.60	92.74	94.17
Mohammed et al. [37]	95.60	93.10	94.35
Deng et al. [21]	96.40	93.30	94.85
Min et al. [56]	95.90	94.70	95.30
Shi et al. [15]	97.00	93.90	95.45
Proposed Framework	97.86	95.36	96.61

4.8. Results on the DHG1428 Dataset

Table 8 presents a detailed evaluation of our framework on the DHG1428 dataset. Utilizing the optimal set of view orientations: custom, top-down, and front-away. Our method achieved validation accuracies of 95.83% for the 14-gesture (14 G) benchmark and 92.38% for the 28-gesture (28 G) benchmark. Compared to the state-of-the-art results reported in [5], our framework performed only slightly lower, with differences of −0.48% (14 G) and −1.67% (28 G).

Table 8. Comparison of Validation Accuracy with SOTA on the DHG1428 Dataset.

Method	Classification Accuracy (%)		
	14 G	**28 G**	**Average**
Lai et al. [57]	85.46	74.19	79.83
Chen et al. [58]	84.68	80.32	82.50
Weng et al. [59]	85.80	80.20	83.00
Devineau et al. [22]	91.28	84.35	87.82
Nguyen et al. [26]	92.38	86.31	89.35
Chen et al. [34]	91.90	88.00	89.95
Mohammed et al. [37]	91.64	89.46	90.55
Liu et al. [6]	92.54	88.86	90.70
Liu et al. [14]	92.71	89.15	90.93
Nguyen et al. [60]	94.29	89.40	91.85
Shi et al. [15]	93.80	90.90	92.35
Proposed Framework	95.83	92.38	94.11
Li et al. [5]	96.31	94.05	95.18

It is worth noting that the DHG1428 dataset exhibits a more balanced distribution of subjects across gesture classes compared to the SHREC2017 dataset, which typically yields somewhat lower accuracy metrics. In alignment with this trend, our framework's performance on DHG1428 was −2.03% and −2.98% lower for the 14 G and 28 G tasks, respectively, compared to its results on SHREC2017 [6,14,15,34].

Figure 6 displays the confusion matrix for the 28 G task, highlighting strong agreement between predicted and actual gesture classes, as evidenced by the robust validation accuracy of 92.38%. In the figure, class labels are prefixed with numbers to distinguish between single-finger gestures (01–14) and whole-hand gestures (15–28).

Consistent with previous findings [11], our analysis reveals a persistent challenge in distinguishing between the "Grab" and "Pinch" gesture classes. This difficulty is present in both performance modes and becomes apparent upon visual inspection of the generated images for these gestures. The pronounced visual similarity between these classes continues to confuse both the model and human evaluators.

4.9. Ablation Study on the SBUKID

Given the structure of our data-level fusion process, we posit that the proposed framework is domain-agnostic and can be readily adapted to other contexts involving the classification of temporal dynamic data represented as 2D or 3D coordinates. To test the generalizability of our framework, we applied it to the skeleton-based Human Action Recognition (HAR) domain, introducing minor modifications to the fusion pipeline. Like hand gesture recognition, HAR tasks require computers to recognize and interpret dynamic actions, but do so by analyzing the motion of the entire human body.

For this evaluation, we used the SBUKID to generate spatiotemporal datasets from all six view orientations. Each of these datasets was used individually to train a custom single-stream CNN model that employed a pre-trained ResNet-50 encoder. As shown in Table 9, our framework achieved an average cross-validation classification accuracy of

93.96% using only the [front-away] view orientation. Although this does not leverage the full e2eET multi-stream CNN architecture, the resulting accuracy is just 4.34% below the state-of-the-art, thereby demonstrating the efficacy of our data-level fusion approach for transforming temporal dynamic data from diverse domains into a format amenable to image classification.

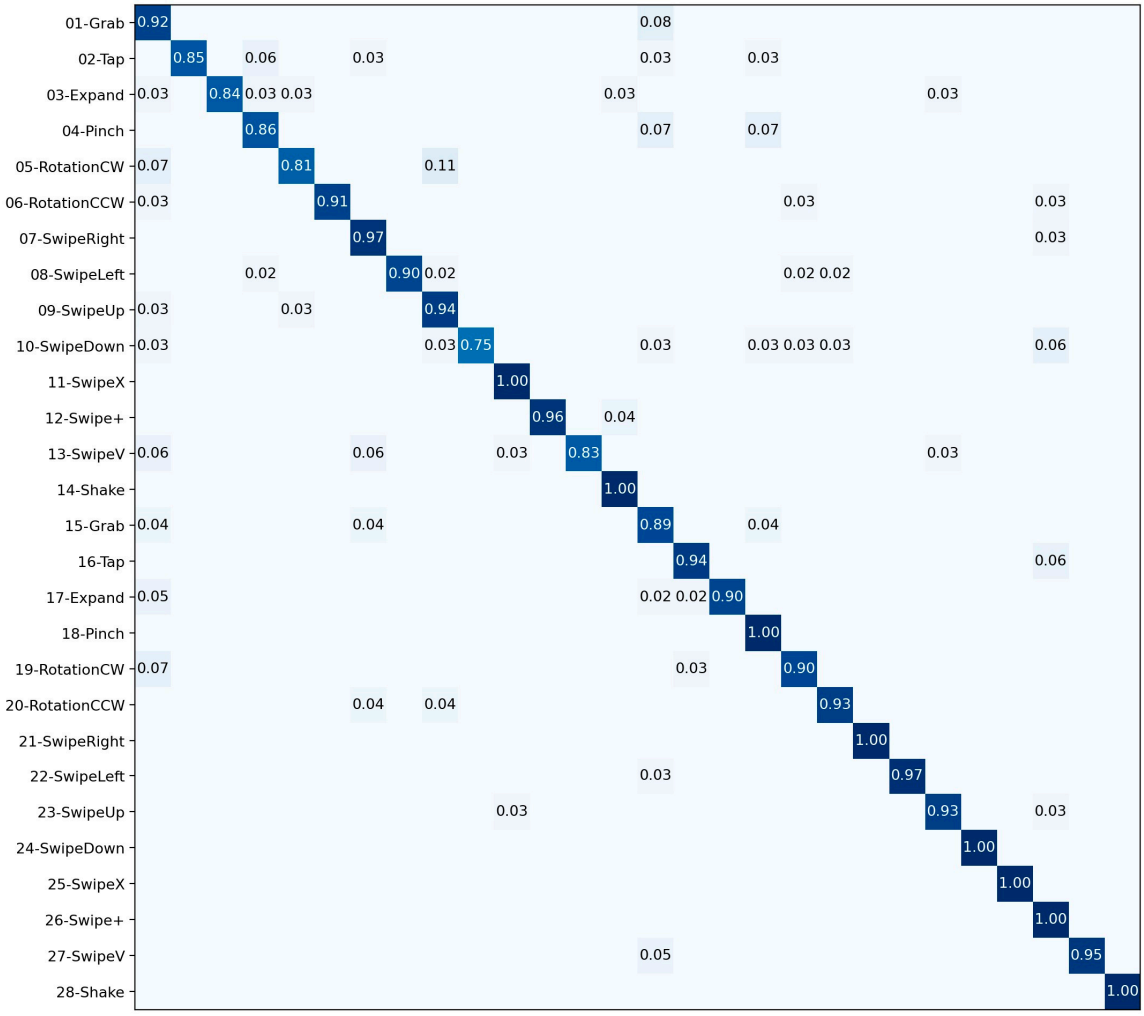


Figure 6. The confusion matrix of the proposed framework on the DHG1428 28 G dataset.

Table 9. Comparison of Validation Accuracy with SOTA on the SBUKID.

Method	Average Cross-Validation Classification Accuracy (%)
Song et al. [61]	91.50
Liu et al. [62]	93.50
Kacem et al. [13]	93.70
Proposed Framework	93.96
Ke et al. [63]	94.17
Mucha & Kampel [64]	94.90
Maghoumi et al. [38]	95.70
Zhang et al. [19]	98.30
Lupinetti, et al. [54]	98.33

To better contextualize the results in Table 9, it is essential to note that the proposed framework was initially designed for hand gesture recognition (HGR), and this HAR evaluation serves as an ablation study to test its cross-domain adaptability with minimal modification. Despite not leveraging temporal modeling layers (e.g., RNNs, LSTMs, or

attention), the method achieves 93.96% accuracy on SBUKID using only static RGB images derived from skeleton sequences and a single-view ResNet-50 encoder. This result demonstrates that the data-level fusion pipeline retains strong semantic structure even outside its native domain. While four existing HAR-specific methods slightly outperform our result, the proposed framework offers several advantages: (i) architectural modularity and portability, (ii) low computational overhead suitable for real-time applications, and (iii) flexibility to integrate additional temporal or view-based modules when needed. Thus, Table 9 reflects a trade-off between domain-specific tuning and generalized adaptability, underscoring the versatility and efficiency of the proposed method.

5. Real-Time HGR Application

To demonstrate the practical utility of our proposed framework in reducing hardware and computational demands for HGR-based applications, we developed a real-time HGR application [10] based on our generalized HGR framework. This application's underlying model was explicitly trained on the "Swipe" gestures from the DHG1428 dataset, including gestures like "Up", "Down", "Right", "Left", "+", "V", and "X". The operational pipeline of the application is depicted in Figure 7 and can be summarized as follows:

(1) OpenCV captures the raw gesture data as RGB videos from the built-in PC webcam.
(2) The video frames are processed for hand detection and pose estimation using MediaPipe Hands.
(3) Data-level fusion is performed on the skeleton data, resulting in three spatiotemporal images from [custom, top-down, front-away] view orientations.

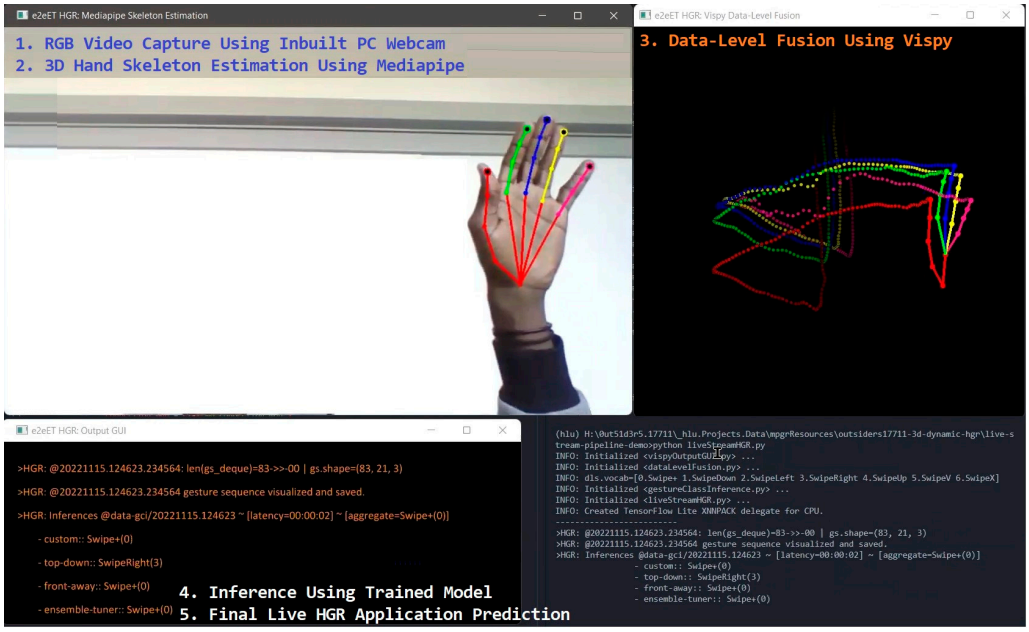


Figure 7. A real-time HGR application was developed, based on our proposed framework for dynamic gesture recognition and classification.

Demonstration of HRG Real-Time Application

To showcase the practical benefits of our framework, particularly its ability to reduce hardware and computational requirements for hand gesture recognition (HGR) applications, we developed a real-time HGR system [10] utilizing our generalized approach. The core model powering this application was trained exclusively on "Swipe" gestures from the DHG1428 dataset, which includes movements such as "Up," "Down," "Right," "Left," "+,"

"V," and "X." The main steps in the system's operational pipeline, illustrated in Figure 7, are as follows:

1. Data Acquisition. OpenCV captures live gesture data as RGB video streams from the built-in PC webcam.
2. Hand Pose Estimation. Video frames are processed with MediaPipe Hands to detect the hand and estimate its pose.
3. Data-Level Fusion. The skeleton data from the detected hand is used to generate three spatiotemporal images corresponding to the [custom, top-down, front-away] view orientations.
4. Gesture Classification. These fused images are input into the trained DHG1428 model, which provides four class predictions—three from the multi-stream sub-network and one from the ensemble tuner sub-network.
5. Results Display. The application's graphical user interface (GUI) presents the predicted gesture classes, relevant details about the gesture sequence, and the framework's end-to-end latency.

The real-time system demonstrates that raw dynamic gesture data can be efficiently collected using a standard PC webcam, eliminating the need for specialized or expensive hardware. On a typical PC equipped with an Intel Core i7-9750H CPU and 16 GB of RAM, the application consistently maintained a latency of just 2–3 s and a capture rate of 15 frames per second, measured from gesture execution to final model prediction. Resource usage remained within acceptable limits, with negligible impact on overall system performance and concurrent applications.

It is essential to clarify that the 2–3 s latency refers to the end-to-end gesture recognition pipeline, encompassing the entire process from the user's completion of a dynamic gesture to the final prediction and display. This includes the processes of skeleton extraction, data-level fusion, multi-stream classification, ensemble tuning, and GUI rendering. The latency does not affect the real-time capture frame rate, which remains at approximately 15 FPS during continuous video acquisition. This prototype was implemented using a modular multi-window Python interface, which contributes additional overhead for I/O and display. A future, integrated, or compiled implementation would likely further reduce latency while preserving model accuracy.

While our real-time prototype was trained on the DHG1428 dataset and achieved consistent performance, the lower accuracy on the CNR dataset is attributed to its lack of raw skeleton data and limited dynamic sequence clarity. Nonetheless, this outcome does not invalidate the model's adaptability. Instead, it highlights the architecture's ability to operate under constrained input scenarios. The ensemble tuning mechanism and multi-view image fusion modules were designed to mitigate such deficiencies, and future training with raw skeleton streams, when available, would further elevate cross-dataset generalizability.

6. Conclusions and Future Work

6.1. Conclusions

This study has examined a range of dynamic hand gesture recognition (HGR) frameworks and introduced a robust skeleton-based HGR approach that streamlines dynamic gesture recognition by reframing it as a static image classification task. The proposed framework preserves essential semantic information by employing an enhanced data-level fusion method to generate static RGB spatiotemporal images from hand pose data. Additionally, the introduction of an ensemble tuner multi-stream CNN architecture, leveraging multiple view orientations, ensures accurate classification of these static images while maintaining computational efficiency.

Comprehensive experiments across five benchmark datasets confirm the effectiveness and generalizability of the framework. The proposed approach achieved validation accuracies ranging from −4.10% below to +6.86% above current state-of-the-art results. Ablation studies in the human action recognition (HAR) domain further demonstrated its versatility, with results only 4.34% shy of the best reported validation accuracy. Collectively, these findings illustrate the framework's ability to handle temporal dynamic data across different contexts.

The framework's real-world applicability was further validated through the development of a real-time HGR application. Using only a standard built-in PC webcam, the application demonstrated efficient performance, maintaining low CPU and RAM usage while delivering acceptable latency and frame rates. This outcome highlights the potential of data-level fusion to reduce hardware and computational requirements without sacrificing system responsiveness.

While the proposed method achieved overall competitive results, performance gaps were observed on datasets such as CNR, FPHA, and DHG1428. These datasets are characterized by challenges such as limited training size, high inter-class similarity (e.g., between gestures like "Grab" and "Pinch"), and significant intra-class variability. These factors can hinder discriminative capacity, mainly when relying solely on skeleton-based static image representations. Future extensions may incorporate attention mechanisms to enhance sensitivity to temporal and joint-specific variations, or apply domain adaptation strategies to improve generalization under such conditions.

While the framework demonstrates strong performance under controlled benchmark conditions, its robustness under variable acquisition scenarios, such as changes in lighting, partial occlusions, or user-specific movement patterns, has not been explicitly assessed. Addressing these variations through perturbation studies and real-world deployment evaluations represents a critical next step to ensure broader applicability and resilience.

Furthermore, the method's computational efficiency stems from its design choices: by reframing gesture recognition as a static classification problem and using lightweight CNN backbones, the framework avoids the overhead associated with temporal sequence models such as RNNs or Transformers. The absence of recurrent computations, combined with compact model architectures and a streamlined preprocessing pipeline, enables faster inference with reduced memory and hardware demands, making it suitable for real-time use cases on standard consumer-grade devices.

6.2. Future Work

Building on the achievements of this work, several promising directions for future research emerge. One natural extension is to adapt and evaluate the framework in the broader context of skeleton-based Human Action Recognition (HAR), capitalizing on the methodological parallels between HGR and HAR for recognizing and classifying human activities from skeletal data.

Further architectural enhancements are also worth pursuing. Incorporating attention mechanisms or transformer-based components into the multi-stream network architecture could yield significant performance improvements, in line with recent advances in deep learning. Investigating the benefits of such enhancements for both hand gesture and broader action recognition tasks represents a fruitful avenue for exploration.

Evaluating the framework in real-world deployment scenarios, such as healthcare or virtual reality, will be essential to assess its practical impact and reveal potential limitations. Such field tests can guide further refinements and adaptation to diverse use cases. Extending the current real-time HGR application and exploring its integration in these domains will provide valuable, actionable insights.

Given the focus on computational efficiency, ongoing research into advanced optimization techniques tailored to the framework's requirements will be essential for maximizing both performance and resource utilization. In addition, conducting thorough user experience (UX) studies will help ensure that the framework remains practical and user-friendly. User feedback gathered from targeted UX evaluations can shape future iterations, promoting continued improvement in usability and effectiveness.

To overcome the current limitation of requiring a fixed user–camera distance during acquisition, future versions of the system could incorporate depth-aware sensors or stereo vision setups. For instance, integrating consumer-grade depth cameras (e.g., Intel RealSense, Azure Kinect) would enable scale-invariant gesture representation and support real-time depth adaptation. This addition could mitigate challenges posed by variable spatial positioning, enhancing the robustness and flexibility of the system across real-world usage scenarios.

While the current study reports classification accuracy as the primary metric, future evaluations will adopt a more comprehensive suite of metrics. Specifically, we will include class-wise precision, recall, and F1-score, which are crucial for handling class imbalance and evaluating classifier discrimination. Additionally, we will implement latency distribution profiling across the end-to-end pipeline, from pose estimation to classification and output rendering, to quantify real-time responsiveness under varying conditions. These enhancements will strengthen the robustness assessment of the proposed framework and support broader deployment readiness.

By pursuing these directions, future research can continue to advance the state of dynamic hand gesture recognition, while deepening understanding of its practical deployment, optimization, and user-centered design.

Author Contributions: Conceptualization, M.K.H., O.Y. and M.M.; methodology, O.Y., M.K.H. and M.M.; software, O.Y.; validation, O.Y., M.K.H. and M.M.; formal analysis, O.Y.; investigation, O.Y., M.K.H. and M.M. writing—original draft preparation, O.Y. and M.K.H.; writing—review and editing, M.K.H.; visualization, O.Y.; supervision, M.K.H. and M.M.; project administration, M.K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: No collection of data from human is taken and the experiment to demonstrate the work done through ion of the authors of this work with his consent according to the AUC institutional Review Board (IRB) requirements.

Data Availability Statement: It is available at https://github.com/Outsiders17711/e2eET-Skeleton-Based-HGR-Using-Data-Level-Fusion?tab=readme-ov-file.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Morris, M.R. AI and Accessibility. *Commun. ACM* **2020**, *63*, 35–37. [CrossRef]
2. Benitez-Garcia, G.; Olivares-Mercado, J.; Sanchez-Perez, G.; Yanai, K. IPN Hand: A Video Dataset and Benchmark for Real-Time Continuous Hand Gesture Recognition. *arXiv* **2020**, arXiv:2005.02134. [CrossRef]
3. Gammulle, H.; Denman, S.; Sridharan, S.; Fookes, C. TMMF: Temporal Multi-Modal Fusion for Single-Stage Continuous Gesture Recognition. *IEEE Trans. Image Process.* **2021**, *30*, 7689–7701. [CrossRef]
4. Lai, K.; Yanushkevich, S. An Ensemble of Knowledge Sharing Models for Dynamic Hand Gesture Recognition. *arXiv* **2020**, arXiv:2008.05732. [CrossRef]
5. Li, C.; Li, S.; Gao, Y.; Zhang, X.; Li, W. A Two-stream Neural Network for Pose-based Hand Gesture Recognition. *arXiv* **2021**, arXiv:2101.08926. [CrossRef]

6. Liu, J.; Liu, Y.; Wang, Y.; Prinet, V.; Xiang, S.; Pan, C. Decoupled Representation Learning for Skeleton-Based Gesture Recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5751–5760. Available online: https://openaccess.thecvf.com/content_CVPR_2020/html/Liu_Decoupled_Representation_Learning_for_Skeleton-Based_Gesture_Recognition_CVPR_2020_paper.html (accessed on 14 October 2025).
7. Wang, Z.; She, Q.; Chalasani, T.; Smolic, A. CatNet: Class Incremental 3D ConvNets for Lifelong Egocentric Gesture Recognition. *arXiv* **2020**, arXiv:2004.09215.
8. Yang, F.; Sakti, S.; Wu, Y.; Nakamura, S. Make Skeleton-based Action Recognition Model Smaller, Faster and Better. *arXiv* **2020**, arXiv:1907.09658. [CrossRef]
9. Zhang, C.; Zou, Y.; Chen, G.; Gan, L. PAN: Towards Fast Action Recognition via Learning Persistence of Appearance. *arXiv* **2020**, arXiv:2008.03462.
10. Yusuf, O.; Habib, M. Development of a Lightweight Real-Time Application for Dynamic Hand Gesture Recognition. In Proceedings of the 2023 IEEE International Conference on Mechatronics and Automation (ICMA), Harbin, China, 6–9 August 2023; pp. 543–548. [CrossRef]
11. Chen, J.; Zhao, C.; Wang, Q.; Meng, H. HMANet: Hyperbolic Manifold Aware Network for SkeletonBased Action Recognition. *IEEE Trans. Cogn. Dev. Syst.* **2023**, *15*, 602–614. [CrossRef]
12. De Smedt, Q.; Wannous, H.; Vandeborre, J.-P.; Guerry, J.; Le Saux, B.; Filliat, D. 3D Hand Gesture Recognition Using a Depth and Skeletal Dataset. In Proceedings of the 3Dor '17: Proceedings of the Workshop on 3D Object Retrieval, Lyon, France, 23–24 April 2017; Eurographics Association: Goslar, Germany, 2017; pp. 33–38. [CrossRef]
13. Kacem, A.; Daoudi, M.; Amor, B.B.; Berretti, S.; Alvarez-Paiva, J.C. A Novel Geometric Framework on Gram Matrix Trajectories for Human Behavior Understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 1–14. [CrossRef]
14. Liu, J.; Wang, G.; Duan, L.Y.; Abdiyeva, K.; Kot, A.C. Skeleton-Based Human Action Recognition with Global Context-Aware Attention LSTM Networks. *IEEE Trans. Image Process.* **2018**, *27*, 1586–1599. [CrossRef]
15. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Decoupled Spatial-Temporal Attention Network for Skeleton-Based Action Recognition. *arXiv* **2020**, arXiv:2007.03263.
16. Gao, S.; Zhang, D.; Tang, Z.; Wang, H. Deep fusion of skeleton spatial–temporal and dynamic information for action recognition. *Sensors* **2024**, *24*, 7609. [CrossRef]
17. Yin, R.; Yin, J. A Two-Stream Hybrid CNN-Transformer Network for Skeleton-Based Human Interaction Recognition. In *Proceedings of the Pattern Recognition and Computer Vision, Proceedings of the 7th Chinese Conference (PRCV 2024), Urumqi, China, 18–20 October 2024; PRCV 2024, Part VII.*; Springer: Berlin/Heidelberg, Germany, 2024; Volume 15037, pp. 395–408. [CrossRef]
18. Molchanov, P.; Gupta, S.; Kim, K.; Kautz, J. Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015; pp. 1–7. [CrossRef]
19. Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; Zheng, N. View Adaptive Neural Networks for High Performance Skeleton-Based Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1963–1978. [CrossRef]
20. Zhang, X.; Li, S.; Zeng, X.; Lu, P.; Sun, W. A Novel Multimodal Hand Gesture Recognition Model Using Combined Approach of Inter-Frame Motion and Shared Attention Weights. *Computers* **2025**, *14*, 432. [CrossRef]
21. Deng, Z.; Gao, Q.; Ju, Z.; Yu, X. Skeleton-Based Multifeatures and Multistream Network for Real-Time Action Recognition. *IEEE Sens. J.* **2023**, *23*, 7397–7409. [CrossRef]
22. Akremi, M.S.; Slama, R.; Tabia, H. SPD Siamese Neural Network for Skeleton-based Hand Gesture Recognition. In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)—Volume 4: VISAPP, Online, 6–8 February 2022; SciTePress: Setúbal, Portugal, 2022; pp. 394–402. [CrossRef]
23. Dang, T.L.; Pham, T.H.; Dao, D.M.; Nguyen, H.V.; Dang, Q.M.; Nguyen, B.T.; Monet, N. DATE: A video dataset and benchmark for dynamic hand gesture recognition. *Neural Comput. Appl.* **2024**, *36*, 17311–17325. [CrossRef]
24. Neverova, N.; Wolf, C.; Taylor, G.W.; Nebout, F. Multi-scale Deep Learning for Gesture Detection and Localization. In *Computer Vision—ECCV 2014 Workshops*; Agapito, L., Bronstein, M., Rother, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 8925. [CrossRef]
25. Liu, X.; Shi, H.; Hong, X.; Chen, H.; Tao, D.; Zhao, G. 3D Skeletal Gesture Recognition via Hidden States Exploration. *IEEE Trans. Image Process.* **2020**, *29*, 4583–4597. [CrossRef] [PubMed]
26. Nguyen, X.S.; Brun, L.; Lézoray, O.; Bougleux, S. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. *arXiv* **2019**, arXiv:1904.12970. [CrossRef]
27. Liu, Y.; Jiao, J. Fusing Skeleton-Based Scene Flow for Gesture Recognition on Point Clouds. *Electronics* **2025**, *14*, 567. [CrossRef]
28. Akrem, M.S. Manifold-Based Approaches for Action and Gesture Recognition/Approches Basées sur les Variétés Pour la Reconnaissance des Actions et des Gestes. Ph.D. Dissertation, IBISC Lab, Paris-Saclay, Paris, France, 2025.
29. Sahbi, H. Skeleton-based Hand-Gesture Recognition with Lightweight Graph Convolutional Networks. *arXiv* **2021**, arXiv:2104.04255.

30. Rehan, M.; Wannous, H.; Alkheir, J.; Aboukassem, K. Learning Co-occurrence Features Across Spatial and Temporal Domains for Hand Gesture Recognition. In Proceedings of the 19th International Conference on Content-based Multimedia Indexing (CBMI '22), Graz, Austria, 14–16 September 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 36–42. [CrossRef]
31. Hu, J.; Wu, C.; Xu, T.; Wu, X.-J.; Kittler, J. Spatio-Temporal Domain-Aware Network for Skeleton-Based Action Representation Learning. In Proceedings of the Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, 1–5 December 2024; Proceedings, Part XXIX. Springer: Cham, Switzerland, 2024; pp. 148–163. [CrossRef]
32. Sabater, A.; Alonso, I.; Montesano, L.; Murillo, A.C. Domain and View-point Agnostic Hand Action Recognition. *arXiv* **2021**, arXiv:2103.02303. [CrossRef]
33. Narayan, S.; Mazumdar, A.P.; Vipparthi, S.K. SBI-DHGR: Skeleton-based intelligent dynamic hand gestures recognition. *Expert Syst. Appl.* **2023**, *232*, 120735. [CrossRef]
34. Chen, Y.; Zhao, L.; Peng, X.; Yuan, J.; Metaxas, D.N. Construct Dynamic Graphs for Hand Gesture Recognition via Spatial-Temporal Attention. *arXiv* **2019**, arXiv:1907.08871. [CrossRef]
35. Mohammed, A.A.Q.; Gao, Y.; Ji, Z.; Lv, J.; Islam, S.; Sang, Y. Automatic 3DSkeleton-based Dynamic Hand Gesture Recognition Using Multi-Layer Convolutional, L.S.T.M. In Proceedings of the 7th International Conference on Robotics and Artificial Intelligence (ICRAI'21), Guangzhou, China, 19–22 November 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 8–14. [CrossRef]
36. Shanmugam, S.; A, L.S.; Dhanasekaran, P.; Mahalakshmi, P.; Sharmila, A. Hand Gesture Recognition using Convolutional Neural Network. In Proceedings of the 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 27–29 November 2021; pp. 1–5. [CrossRef]
37. Min, Y.; Zhang, Y.; Chai, X.; Chen, X. An Efficient PointLSTM for Point Clouds Based Gesture Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5761–5770. Available online: https://openaccess.thecvf.com/content_CVPR_2020/html/Min_An_Efficient_PointLSTM_for_Point_Clouds_Based_Gesture_Recognition_CVPR_2020_paper.html (accessed on 14 October 2025).
38. Maghoumi, M.; LaViola, J.J., Jr. DeepGRU: Deep Gesture Recognition Utility. *arXiv* **2019**, arXiv:1810.12514. [CrossRef]
39. Song, J.-H.; Kong, K.; Kang, S.-J. Dynamic Hand Gesture Recognition Using Improved Spatio-Temporal Graph Convolutional Network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6227–6239. [CrossRef]
40. Molchanov, P.; Yang, X.; Gupta, S.; Kim, K.; Tyree, S.; Kautz, J. Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4207–4215. [CrossRef]
41. Köpüklü, O.; Köse, N.; Rigoll, G. Motion Fused Frames: Data Level Fusion Strategy for Hand Gesture Recognition. *arXiv* **2018**, arXiv:1804.07187. [CrossRef]
42. Zhou, H.; Le, H.T.; Zhang, S.; Phung, S.L.; Alici, G. Hand Gesture Recognition from Surface Electromyography Signals with Graph Convolutional Network and Attention Mechanisms. *IEEE Sens. J.* **2025**, *25*, 9081–9092. [CrossRef]
43. Wang, B.; Lu, R.; Zhang, L. Efficient Hand Gesture Recognition Using Multi-Stream CNNs with Feature Fusion Strategies. *IEEE Access* **2021**, *9*, 23567–23578.
44. Abavisani, M.; Joze, H.R.V.; Patel, V.M. Improving the Performance of Unimodal Dynamic Hand-Gesture Recognition with Multimodal Training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1165–1174. Available online: https://openaccess.thecvf.com/content_CVPR_2019/html/Abavisani_Improving_the_Performance_of_Unimodal_Dynamic_Hand-Gesture_Recognition_With_Multimodal_CVPR_2019_paper.html (accessed on 14 October 2025).
45. Liu, W.; Lu, B. Multi-Stream Convolutional Neural Network-Based Wearable, Flexible Bionic Gesture Surface Muscle Feature Extraction and Recognition. *Front. Bioeng. Biotechnol.* **2022**, *10*, 833793. [CrossRef] [PubMed]
46. Yang, X.; Molchanov, P.; Kautz, J. Making Convolutional Networks Recurrent for Visual Sequence Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6469–6478. Available online: https://openaccess.thecvf.com/content_cvpr_2018/html/Yang_Making_Convolutional_Networks_CVPR_2018_paper.html (accessed on 14 October 2025).
47. Yu, Z.; Zhou, B.; Wan, J.; Wang, P.; Chen, H.; Liu, X.; Li, S.Z.; Zhao, G. Searching Multi-Rate and Multi-Modal Temporal Enhanced Networks for Gesture Recognition. *IEEE Trans. Image Process.* **2021**, *30*, 5626–5640. [CrossRef]
48. Li, L.; Qin, S.; Lu, Z.; Zhang, D.; Xu, K.; Hu, Z. Real-time one-shot learning gesture recognition based on lightweight 3D Inception-ResNet with separable convolutions. *Pattern Anal. Appl.* **2021**, *24*, 1173–1192. [CrossRef]
49. Kendall, A.; Gal, Y.; Cipolla, R. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *arXiv* **2018**, arXiv:1705.07115. [CrossRef]

50. Boulahia, S.Y.; Anquetil, E.; Multon, F.; Kulpa, R. Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In Proceedings of the Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–6. [CrossRef]
51. Devineau, G.; Xi, W.; Moutarde, F.; Yang, J. Deep Learning for Hand Gesture Recognition on Skeletal Data. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018. [CrossRef]
52. De Smedt, Q.; Wannous, H.; Vandeborre, J.-P. Skeleton-Based Dynamic Hand Gesture Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1206–1214. [CrossRef]
53. Kim, S.; Jung, J.; Lee, K.J. A Real-Time Sparsity-Aware 3D-CNN Processor for Mobile Hand Gesture Recognition. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2024**, *71*, 3695–3707. [CrossRef]
54. Lupinetti, K.; Ranieri, A.; Giannini, F.; Monti, M. 3D dynamic hand gestures recognition using the Leap Motion sensor and convolutional neural networks. *arXiv* **2020**, arXiv:2003.01450. [CrossRef]
55. Zhang, C.; Zou, Y.; Chen, G.; Gan, L. PAN: Persistent Appearance Network with an Efficient Motion Cue for Fast Action Recognition. In Proceedings of the MM '19: Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 500–509. [CrossRef]
56. Miah, A.S.M.; Hasan, A.M.; Shin, J. Dynamic Hand Gesture Recognition Using Multi-Branch Attention Based Graph and General Deep Learning Model. *IEEE Access* **2023**, *11*, 4703–4716. [CrossRef]
57. Lai, K.; Yanushkevich, S.N. CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 3451–3456. [CrossRef]
58. Chen, X.; Guo, H.; Wang, G.; Zhang, L. Motion Feature Augmented Recurrent Neural Network for Skeletonbased Dynamic Hand Gesture Recognition. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2881–2885. [CrossRef]
59. Weng, J.; Liu, M.; Jiang, X.; Yuan, J. Deformable Pose Traversal Convolution for 3D Action and Gesture Recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 136–152. Available online: https://openaccess.thecvf.com/content_ECCV_2018/html/Junwu_Weng_Deformable_Pose_Traversal_ECCV_2018_paper.html (accessed on 14 October 2025).
60. Nguyen, X.S.; Brun, L.; Lezoray, O.; Bougleux, S. Skeleton-Based Hand Gesture Recognition by Learning SPD Matrices with Neural Networks. *arXiv* **2019**, arXiv:1905.07917. [CrossRef]
61. Song, S.; Lan, C.; Xing, J.; Zeng, W.; Liu, J. Spatio-Temporal Attention-Based LSTM Networks for 3D Action Recognition and Detection. *IEEE Trans. Image Process.* **2018**, *27*, 3459–3471. [CrossRef] [PubMed]
62. Liu, J.; Wang, Y.; Xiang, S.; Pan, C. HAN: An Efficient Hierarchical Self-Attention Network for Skeleton-Based Gesture Recognition. *arXiv* **2021**, arXiv:2106.13391. [CrossRef]
63. Ke, Q.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F. Learning Clip Representations for SkeletonBased 3D Action Recognition. *IEEE Trans. Image Process.* **2018**, *27*, 2842–2855. [CrossRef]
64. Mucha, W.; Kampel, M. Beyond Privacy of Depth Sensors in Active and Assisted Living Devices. In Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments, Corfu, Greece, 29 June–1 July 2022; ACM: Corfu, Greece, 2022; pp. 425–429.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.