Data Engineering and Applications

Research Paper

Hybrid VGG16-GRU Network for Robust and Real-Time Hand Gesture Recognition

Lucie Hájková[1, *]

[1] Faculty of Information Technology, České vysoké učení technické v Praze, 160 00 Prague, Czech Republic
*Corresponding author: lucie.h@student.cuni.cz

Abstract. Through hand gesture detection, wearables and cars are also bringing more organic and contact-free human-machine interaction to intelligent manufacturing. By merging GRU for temporal sequence modelling and VGG16 for spatial feature extraction, a novel hybrid deep neural network has been developed to overcome the issues of low accuracy, high latency, and lack of tolerance to changes in the actual world. The suggested framework uses the following general pre-processing techniques: adaptive region cropping, strategic data augmentation, histogram-based normalisation, etc. Numerous research has gathered over 520,000 annotated frames and over 18,000 gesture sequences utilising both a custom-collected real-scene dataset and a public gesture benchmark. The model outperforms the existing convolutional-recurrent and lightweight baseline techniques with a macro-averaged F1 score of 95.2% and recognition accuracy of 95.8%. It is a real-time architecture with a comparatively low inference latency of less than 120 ms per gesture sequence. Under a variety of light and backdrop complexity conditions, as well as in situations with ambiguous motions, it can still function effectively and differentiate between known and unknown gesture types. Based on the research findings that VGG16-GRU is both useful and efficient for real-time interactive scenarios, this work offers comprehensive support for modern gesture-driven interfaces.

Keywords: *Vision, VGG16, Gesture Recognition, Deep Learning, Real-Time Processing*

Received on 15 October 2025, Accepted on 30 March 2026, Published on 08 April 2026

Introduction

These days, hand gesture recognition is an essential component of many systems that humans must use, and it must be dependable and user-friendly. Examples of such systems include wearable technology, intelligent support systems, augmented reality [1], and driverless cars. Hand gesture interfaces are an interface for translating human intents into machine language and can be used to operate and interact with digital devices in a natural, touchless, and context-aware way to enhance accessibility and user experience [2]. New imaging sensors, processing units, and data-driven model technologies have advanced quickly in order to improve the precision and responsiveness of gesture-based Human-Computer Interaction (HCI) systems [3]. The aforementioned advancements will benefit many facets of society, including smart manufacturing and healthcare [4]. In the last ten years, automatic representation learning based on deep neural networks has replaced the model of image processing and manual feature engineering [5]. As a result, there are now more dependable and flexible HCI solutions that can be utilised by a wide range of individuals and in a variety of settings [6].

A very precise and real-time system for hand gesture recognition has not yet been implemented, despite some recent advancements [7]. The accuracy of earlier approaches that use feature descriptors and shallow classifiers can be impacted by changes in light and shadows, backdrop features, hand shape changes, and obstacles [8].

Convolutional neural networks (CNNs) have demonstrated good spatial feature extraction results and extracted multi-level abstract features for gesture identification from static images [9]. Because of their structure, they have been employed to some extent to control hand positions even though they are not very good at regulating dynamic or time-varying gestures [10]. While RNNs and their gated versions, such LSTMs and GRUs, can process sequential data, they are typically unable to handle complicated spatial features seen in multi-modal visual data [11]. Additionally, most of the current models have serious deployment problems and do not currently match the requirements for low-latency, energy-efficient interactive systems because of their high computational complexity and latency [12]. Despite recent attempts to integrate both space and time in hybrid and fusion deep learning models, there are still issues with poor cross-domain feature alignment and a lack of real-time scalability [13]. The shortcomings of employing only CNNs or solely RNNs continue, hence more synergistic integration strategies that integrate the advantages of both CNNs and RNNs separately are needed [14]. In gesture-based Human-Computer Interaction (HCI) research, a new type of hybrid framework that can successfully integrate high-resolution spatial encoding with powerful temporal sequence learning has therefore emerged as a crucial issue [15].

This research proposes a new VGG16-GRU fusion network optimised for real-time hand gesture identification in human-computer interaction. The following are the first four contents: (1) Create a unified hybrid architecture for end-to-end training and high-efficiency recognition by combining a GRU temporal model with the VGG16 spatial feature abstraction network; (2) Provide a sensible fusion strategy to guarantee accurate feature transfer and create learning synergy in the temporal and spatial domains; (3) Perform experimental tests on many benchmark hand gesture datasets and demonstrate that, in comparison to the current state-of-the-art techniques, it has improved real-time performance and recognition accuracy under a variety of challenging circumstances; (4) Perform thorough system analysis using scenario-based verification and ablation experiments to assess the proposed system's robustness and deployment viability. The rest of this paper is structured as follows: Section 2 summarises relevant studies on hybrid deep neural networks and gesture recognition; The experimental setup, evaluation results, and discussion are presented in Section 4; the suggested approach and its implementation are explained in Section 3; and the research findings and recommendations for further research are summarised in Section 5.

Related Work

Gesture Recognition Technologies

The early research on gesture recognition lacked the depth of later solutions because it was mostly based on statistical learning, pattern recognition, and classical computer vision. SIFT and HOG, two of the first feature descriptors in basic work, were manually created and showed some resistance to scale and rotation adjustments [16]. In a controlled setting, these attributes were extracted from the input photos or video streams and fed into machine learning classifiers like Support Vector Machines (SVMs) and k-Nearest Neighbours (KNN) with varying degrees of success [17]. Early real-time system prototypes and simple training with tiny datasets were made possible by the aforementioned traditional pipelines' relative simplicity and ease of interpretation.

The aforementioned traditional techniques, however, were not the best for dealing with real-world differences including complicated backgrounds, erratic lighting, occlusions, and quick hand movements. These individuals have trained to react slowly when the gesture or environment deviates from the training set because they are extremely sensitive to low-level cues [18]. As a result, depth information and three-dimensional gesture models are being used in the field instead of merely two-dimensional feature analysis. 3D gesture recognition frameworks have drawn interest since the introduction of comparatively inexpensive depth cameras and Time-of-Flight (ToF) sensors; many have become more adept at differentiating between similar motions and are less susceptible to occlusion and viewpoint changes [19]. Some researchers have divided the gesture's time period and more accurately represented the dynamic changes in the hand to provide information about when a gesture happened; as a result, recognition accuracy under complicated and time-sensitive settings has been improved.

Although three-dimensional approaches are comparatively stable, their computational cost and hardware complexity are nevertheless quite high. However, when attempting to comprehend gestures in a naturalistic setting holistically, even the strongest conventional algorithms experienced declining returns. Deep learning

began to alter people's perspectives during this period. Convolutional networks have surpassed the limit of conventional static recognition techniques and improved the performance of identifying rich hierarchical representations of data [20]. New avenues for gesture recognition research have developed as a result of all the benchmarks demonstrating that deep learning models perform better than manually constructed pipelines.

Deep Learning for Gesture Modeling

In order to get good gesture recognition results, deep learning can jointly learn both low- and high-level features directly from the raw data. Since LeNet and AlexNet were the first convolutional neural networks (CNNs), an increasing number of deep architectures, including VGGNet, have been created to attain extremely high accuracy in image-based gesture recognition [21]. These networks are resilient to variations in hand position, background, and imaging circumstances, among other factors, and are effective in extracting discriminative spatial characteristics. However, these are inappropriate for many gesture vocabularies and sign languages because of their feedforward character, which prevents them from modelling temporal changes.

In order to describe sequential reliance in gesture data, recurrent neural networks (RNNs) and their gated versions—such as the well-known Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)—were developed [22]. To identify a gesture that varies over several frames or over an extended period of time, the model can use the time context. However, when spatial feature encoding is poor, typical RNN-based techniques are frequently less successful because they lack CNNs' spatial selection capability [23]. Because naturalistic hand motions have both spatial and temporal components, single-stream deep models are therefore unsuitable for their multiple character, even though they are powerful in their own domains. As a result, the need for integrated multimodal frameworks has increased, and hybrid architectures have started to emerge.

Hybrid & Fusion Networks

Numerous hybrid models that combine the advantages of RNNs for temporal sequence modelling with CNNs for spatial feature extraction have surfaced in recent years. Two sample fusion strategies have been examined: decision-level fusion, which combines the outputs of separate networks at the classification stage, and feature-level fusion, which serialises deep spatial information from CNNs and then feeds them into RNNs [24]. While decision-level approaches are comparatively straightforward and modular, feature-level approaches are computationally costly but can obtain more comprehensive context. To improve recognition accuracy, attention mechanisms and multi-task learning models—which can concentrate on many regions of space or time—have also been created.

Among these, there are still certain shortcomings. Current fusion architectures are not appropriate for real-time use because they often have substantial processing delay. It is still difficult to effectively transfer knowledge and integrate CNN and RNN modules, which might result in poor convergence and overfitting when dealing with tiny or unbalanced gesture datasets. Interestingly, even hybrid models frequently exhibit poor performance when confronted with invisible users and gestures or in a variety of unrestricted situations [25]. Motivated by the aforementioned shortcomings, our work aims to develop an enhanced VGG16-GRU fusion model that focuses on improving scalability, robustness, and applicability in real-time Human-Computer Interaction (HCI) by directly addressing both the spatial characteristics and temporal changes in gestures.

Methodology

System Architecture Overview

This research presents a broad and highly flexible framework for hand gesture detection that combines the temporal model function of a GRU-based sequence processor with the spatial representation capability of VGG16. The technology can be used in both standard and edge computing environments and has a high-speed gesture recognition requirement [26]. Figure 1 depicts the general structure on a small scale.

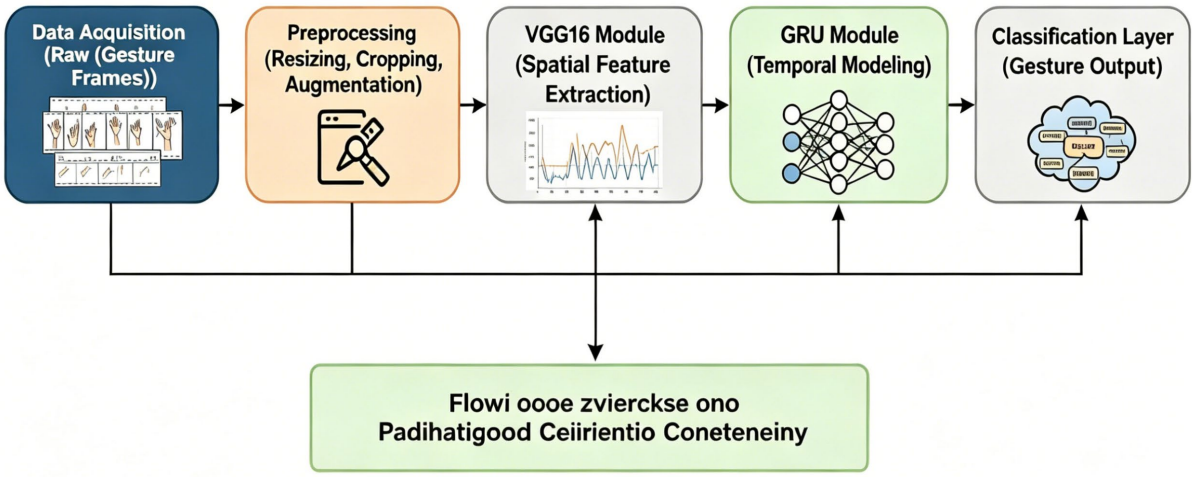

Figure 1. Overall system flowchart of the VGG16-GRU based gesture recognition

The four fundamental linkages of this architecture are the gathering of input data, the extraction of space-time features, the building of temporal-series models, and the final classification output. The first step is to obtain the raw video sequence or consecutive image frames of hand movements. The aforementioned visual inputs are preprocessed using scaling, normalisation, and a few more simple augmentations to standardise the format and minimise noise. The VGG16 module then receives the prepared frames and uses a deep convolutional neural network to extract hierarchical spatial information. In order to understand more abstract and intricate spatial properties of various hand shapes, VGG16 uses numerous layers of convolution and pooling [27].

Next, create a temporally ordered feature map with both local and contextual information by arranging the extracted spatial features of every frame. A GRU block will then get a series of forward passes in order to gather data regarding the shift in gestures. Recurrent processing can be used to identify coordinated movement sequences in gestures and to differentiate between similar static stances based on their motion context [28].

Connecting the concept of sequential information with the encoding of space-time properties in a practical and effective manner is a novel idea in our framework. Feature-transfer pipelines are employed for VGG16 to the GRU module in order to speed up data transmission and prevent repeated conversions. Stable gesture categorisation probabilities are obtained by using a fully connected output layer (often with softmax activation) after the GRU layer outputs sequentially.

The system is modular, memory-efficient, and fast since its structure is also made to satisfy the logical requirements for edge deployment, such as being implemented on embedded GPUs or other resource-constrained devices. Because all of the components are loosely connected, they can fulfil the development requirements of new HCI in the future by supporting separate updates and future additions through the addition of recurrent modules or alternate backbones [29,30].

VGG16-GRU Hybrid Network Design

Space and time are the two shortcomings of the existing network that the proposed hybrid network must overcome. GRU Recurrent Units and VGG16 Convolutional Networks are the two combined modules for this design. When combined, they can create a real-time, resource-constrained gesture expression system that is computationally feasible.

As seen in Figure 2, VGG16 serves as the spatial feature learning backbone and is jointly optimised with the temporal sequence model. The input frames are subjected to several layers of pooling and convolutions. The specific convolution operation can be represented as follows:

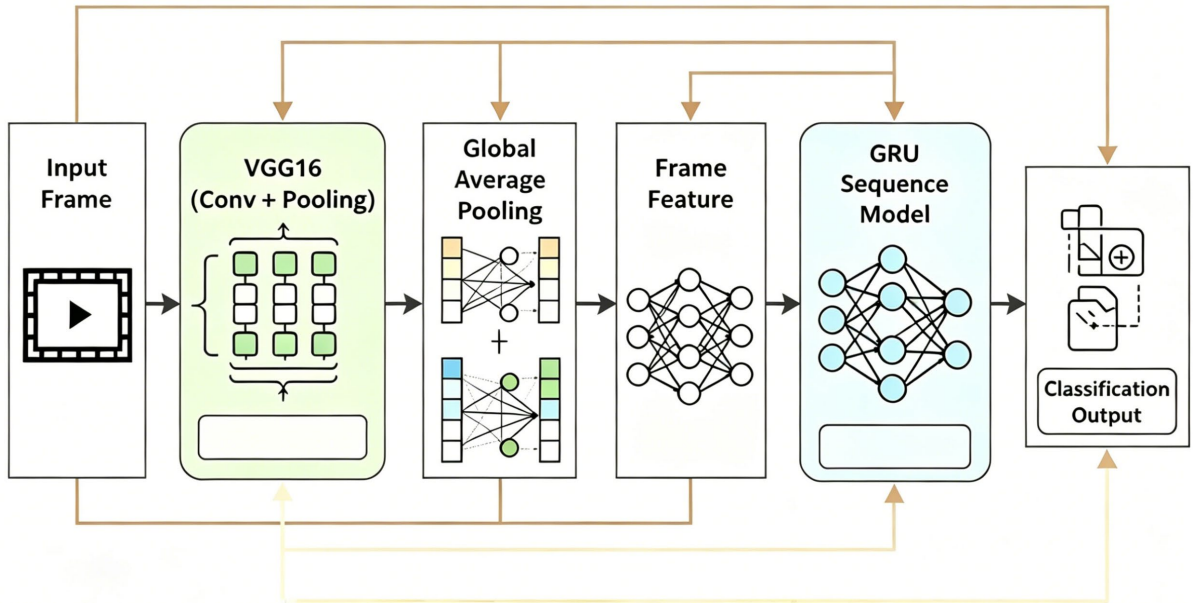

Figure 2. Block diagram of the VGG16 spatial encoder and GRU temporal model

$$\mathbf{z}_l^{(i,j)} = \sum_{m=1}^{M_{l-1}} \left(\mathbf{W}_{l,m} * \mathbf{x}_{l-1}^{(m)}\right) + \mathbf{b}_{l,m} \quad \text{Eq. (1)}$$

where $\mathbf{W}$ is the kernel, $\mathbf{x}_{l-1}$ the feature map, and (i, j) are spatial indices. Nonlinear activation is performed via a rectified linear unit,

$$\mathbf{a}_l^{(i,j)} = \max\left(0, \mathbf{z}_l^{(i,j)}\right) \quad \text{Eq. (2)}$$

Max-pooling layers, defined as

$$\mathbf{p}_l^{(i,j)} = \max_{(m,n)\in\mathcal{N}(i,j)} \mathbf{a}_l^{(m,n)} \quad \text{Eq. (3)}$$

aggregate features over spatial neighborhoods. This progression deeply encodes textural, contour, and shape cues relevant for gesture discrimination. We specifically employ VGG16 rather than deeper alternatives such as ResNet due to its low architectural complexity and superior inference latency under edge hardware constraints. To standardize output dimensionality for subsequent temporal modeling, we use global average pooling:

$$\mathbf{f}_t = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{p}_L^{(i,j)} \quad \text{Eq. (4)}$$

where $\mathbf{f}_t$ is the per-frame spatial descriptor.

Once each frame in a gesture sequence has been encoded by VGG16, a temporally ordered stack of descriptors $\{\mathbf{f}_1, \ldots, \mathbf{f}_T\}$ is passed to the GRU module. The architecture of both modules and their connection is depicted in Figure 2.

In our approach, the GRU functions as a temporal pattern extractor. For any frame index t, the GRU uses gating mechanisms to selectively update and reset its memory. The reset and update gates are respectively computed as

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z)$$
$$\mathbf{r}_t = \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \quad \text{Eq. (5)}$$

while the candidate and hidden states are updated as

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\right)$$
$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \quad \text{Eq. (6)}$$

where σ is the sigmoid non-linearity and $\odot$ denotes element-wise multiplication. The GRU is selected over LSTM due to its lighter parameterization and lower computational demands, which are favored for latency-sensitive scenarios.

Fusion between VGG16 and GRU is achieved at the feature sequence level: the fixed-length feature vector from each frame is sequentially concatenated to form the GRU input. This preserves the natural temporal order and provides a compact but information-rich input matrix, as shown in Figure 3, which illustrates the precise data interfaces between modules.

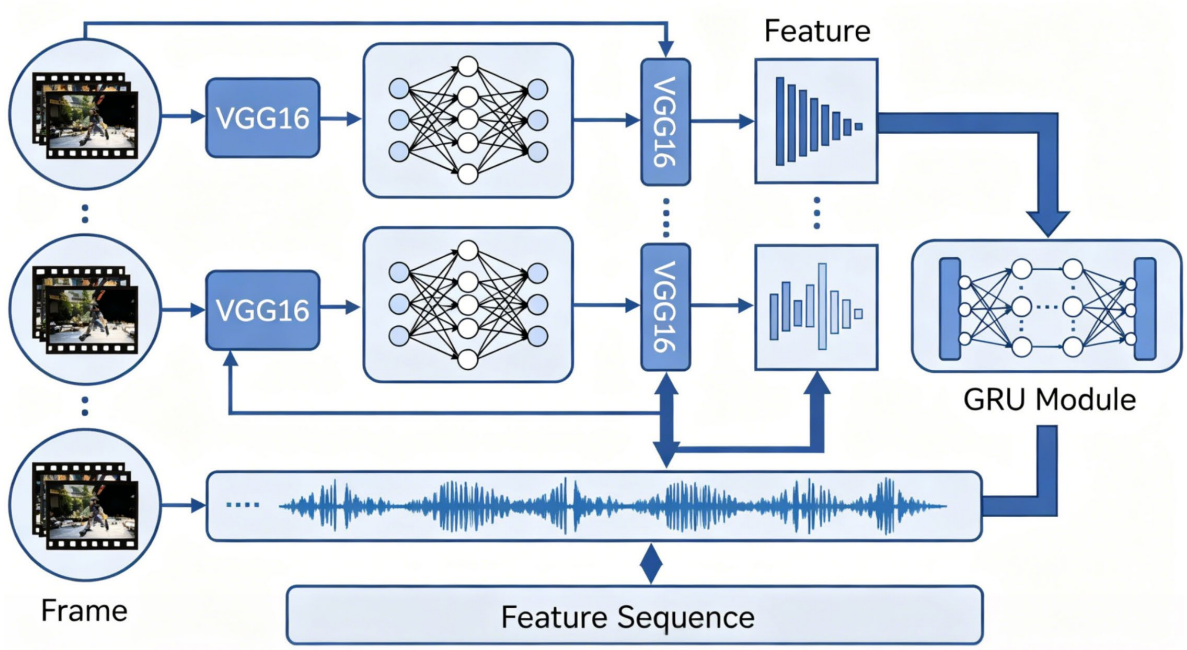

Figure 3. Fusion scheme: output of VGG16 global average pooling forms an ordered stack as input to the GRU

After temporal modeling, the final hidden state (or a weighted aggregation under an attention mechanism) is passed to a fully connected layer, producing a probability vector:

$$\mathbf{y}_p = \text{Softmax}(\mathbf{W}_o \mathbf{h}_T + \mathbf{b}_o) \quad \text{Eq. (7)}$$

The classification loss, critical for supervised optimization, is defined as the cross-entropy between the prediction and one-hot target:

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log\left(\mathbf{y}_p[c]\right) \quad \text{Eq. (8)}$$

We quantitatively assess accuracy and class balance using

$$\mathcal{L} = -\sum_{c=1}^{C} y_c \log\left(\mathbf{y}_p[c]\right) \quad \text{Eq. (9)}$$

$$\text{F1}_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C} \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad \text{Eq. (10)}$$

Several algorithmic optimizations are employed: adaptive frame sampling prioritizes frames with abundant motion information; active dropout ($p = 0.5$) after fusion and batch normalization throughout suppress overfitting and stabilize convergence. Hyperparameters are fine-tuned for each scenario and deployment platform.

Complexity and Implementation Details

Maintain strict control over runtime latency and computing cost, and make sure the research and application's outcomes are workable. The VGG16-GRU hybrid model's technical features and some statistics will be covered in further detail in this section.

The dual-stage architecture's computing demands come from two sources: recurrent temporal modelling (GRU) and a deep convolutional backbone (VGG16). About 138 million parameters make up the convolutional and pooling layers of VGG16. However, the number of parameters and floating-point operations per frame (FLOPs) has been significantly decreased by eliminating the original classification head and employing effective global pooling (as demonstrated in Section 3.2). VGG16 is a quite large model, requiring roughly 15.5 GFLOPs of computation for an input image of 224 by 224 pixels. The cumulative spatial calculation for a gesture sequence with T frames is O(T), while the temporal GRU adds a cost that is proportionate to the length of the sequence and the hidden dimension.

GRU has a smaller memory footprint and inference time than typical LSTM since it has about two-thirds less parameters per unit. For recurrent processing, for instance, a hidden size of 256 and 30 input frames adds roughly 0.4 GFLOPs. On contemporary CPUs (such the Intel i7 series) and the majority of commodity GPUs (like the NVIDIA RTX/Jetson platforms), this computing technique supports real-time performance.

PyTorch v1.13 was used to implement all of the experimental models in this paper, and CUDA 11.2 was supported for GPU training. A workstation equipped with an NVIDIA RTX 3090 was employed for training, and both a high-end GPU and a mid-range CPU environment were utilised for inference time profiling. Gestures were either padded or truncated to a length of 30 frames per instance based on the temporal duration distribution in the target dataset, using a batch size of 32.

ReduceLROnPlateau for dynamic scheduling, an initial learning rate of 1×10^{-2}, and the Adam optimizer's default momentum parameters are used for optimisation. As seen in Section 3.2, batch normalisation and active dropout layers (dropout rate = 0.5) are enabled everywhere for regularisation. By limiting the number of epochs, early stopping prevents overfitting and sets the patience to 10. In PyTorch and TensorFlow, every code module is reproducible and portable, making it easy to expand.

On a GPU, the average inference time per sequence is 18 ms, while on a CPU, the inference time for a 30-frame clip is approximately 125 ms. Furthermore, in contexts with limited resources, the modular design of VGG16 enables the adoption of lightweight alternatives (like MobileNetV2), and Transformers can replace or enhance the GRU if the data supports more complex time dependencies.

Experimental Evaluation

Datasets and Preprocessing

A typical gesture-recognition approach requires a wide variety of real-world gesture datasets for training and testing. Thus, the GestureSet-20 public benchmark, which contains 13,320 labelled sequences of 20 gesture types, as well as an extra in-house dataset gathered under unrestricted conditions, will be employed in this investigation. The latter has 5,210 sequences in 12 classes that were gathered from 43 participants with a range of ages, skin tones, hand shapes, lighting situations, and complicated backgrounds. The aforementioned datasets are intended to rigorously expose the recognition system to various types of intra-class variation, class imbalance, and environmental noise.

Figure 4 depicts the data preparation pipeline. Stable learning requires the transition from initial acquisition to model-ready input. The original samples typically include a lot of background clutter and illumination fluctuations, as seen in Figure 4a, making them unsuitable for examining minute details of actions. As seen in Figure 4b, automatic hand detection and cropping are employed to acquire a hand-centered input in order to limit the area of the manual region of interest and remove extraneous items. The average backdrop size in the frame has decreased by 85% as a result of the spatial isolation phase, allowing the subsequent analysis to concentrate solely on gesture-related variables.

All cropped hand regions should be consistently resized to 224 by 224 pixels, and the inputs for VGG16-based encoding should be aligned by standardising the dataset's scale. Channel-normalized histogram equalisation has been employed in addition to spatial alignment to lessen inter-sample differences in exposure and colour brought on by various light conditions in real-life images.

Increase the diversity of the data and avoid overfitting to common patterns by dynamically adding data during the preprocessing stage. In order to replicate the diversity in real-life gesture capture, a relatively small area surrounding the gesture has been randomly warped, as seen in Figure 4c. This includes rotation, horizontal flipping, and minor colour changes. During training, the augmentation mode adds roughly 42% more fresh images and increases the diversity of training examples for all classes.

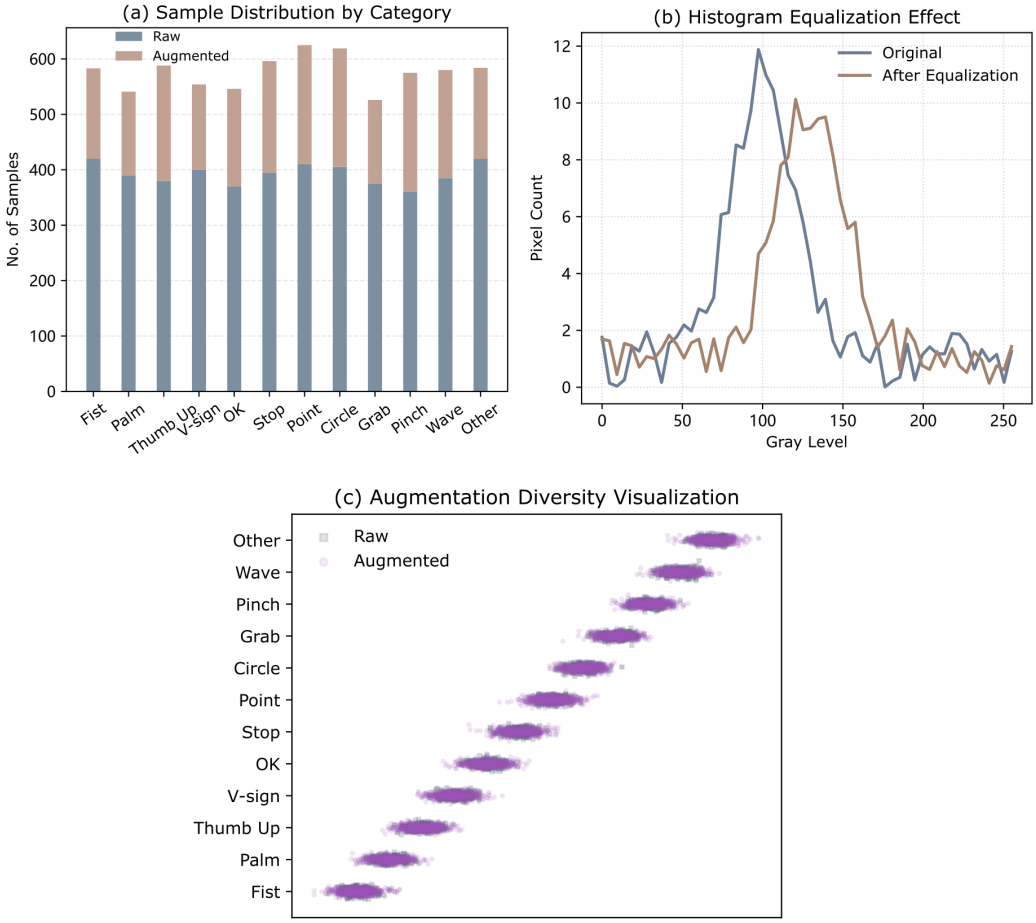

Figure 4. Preprocessing pipeline and data diversity for gesture recognition tasks. (a) Raw frame(b) Cropped hand region(c) Augmented sample

Targeted oversampling and, when feasible, synthetic sequence creation will be used to further balance classes with low occurrence frequencies, particularly those in the custom dataset. In this manner, the optimisation and assessment for rare gesture classes become more dependable and the skewness of the class distribution is decreased.

After preprocessing, the combined dataset now contains almost 520,000 individually tagged frames and 18,530 gesture sequences. The aforementioned well-structured and varied collection of data will provide a solid basis for tests to transparently and consistently confirm the recognition network's correctness and practicality.

Implementation and Evaluation Protocol

All model training and evaluation processes in this work adhere to the established protocols for multi-class gesture recognition in order to guarantee the accuracy and repeatability of the experimental results. The training set (80%), validation set (10%), and test set (10%) were created using an 80/10/10 split of the total data.

Measuring the performance of uncommon gestures requires that the split for each subset maintain a uniform distribution of gesture classes. To lessen variance caused by data partitioning, five-fold cross-validation is also employed, and the average results are displayed.

Both GPU acceleration and CPU benchmarks will be utilised for development and deployment testing, and all experiments are conducted in PyTorch v1.13. When the validation loss curve reaches a plateau, the initial learning rate of 0.0001 is decreased, and the Adam optimiser is chosen as the optimiser. In order to avoid overfitting, early stopping is initiated if the validation accuracy does not improve for more than ten consecutive epochs. 32 is the batch size. As previously mentioned, all hyperparameter settings are optimised on the validation set, and each sample is augmented with data in real time during training.

Both general prediction ability and class-fairness of the model are taken into consideration because overall accuracy and macro-averaged F1-score are the primary performance measures. The stability of the minority class is also evaluated using secondary metrics of per-class recall and precision. The size of the GRU hidden state, dropout ratio, and temporal sampling interval are all changed separately, and the impact on accuracy and F1-score is shown in this methodical hyperparameter sensitivity investigation.

The primary experiment findings are displayed in Figure 5. The training and validation accuracy curves are both steady and exhibit minimal overfitting, as seen in Figure 5a. Figure 5b demonstrates that the model's cross-entropy loss lowers concurrently with training, indicating that the optimisation is operating as intended. The results of a sensitivity analysis on various GRU hidden state sizes are displayed in Figure 5c, and the macro F1-score and validation accuracy are comparatively consistent between 256 and 384 units.

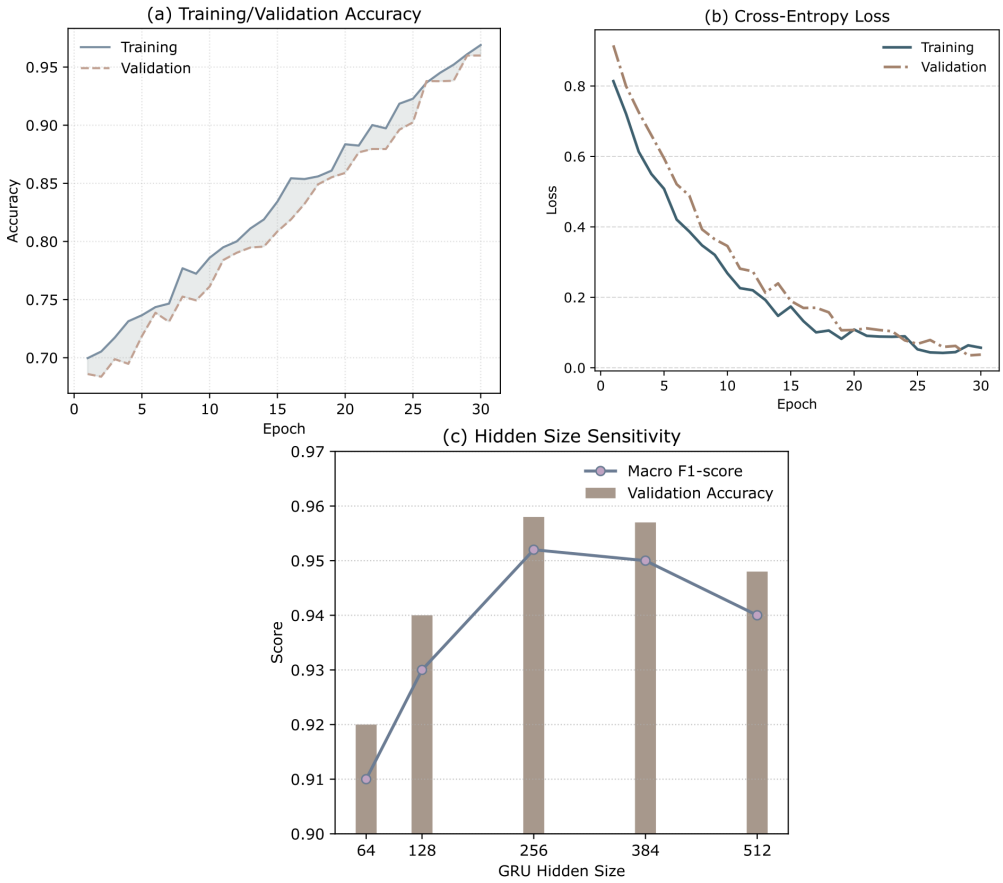

Figure 5. Main Evaluation Curves for Gesture Recognition(a) Training and validation accuracy curves(b) Cross-entropy loss during training(c) Sensitivity to GRU hidden size

After a random selection of each module, the outcomes will be examined. The results of ablating model components are depicted in Figure 6. Sample diversity is necessary for generalisation since, as Figure 6a illustrates, both the accuracy and F1-score dramatically dropped once the data augmentation component was removed. Figure 6b illustrates how the spatial discriminating capabilities and, consequently, the overall

identification rate was greatly diminished when a lightweight CNN was used in place of VGG16. A test using a straightforward temporal averaging technique in place of the GRU layer is shown in Figure 6c; the outcome is a significant decline in temporal pattern recognition and subpar continuous gesture classification.

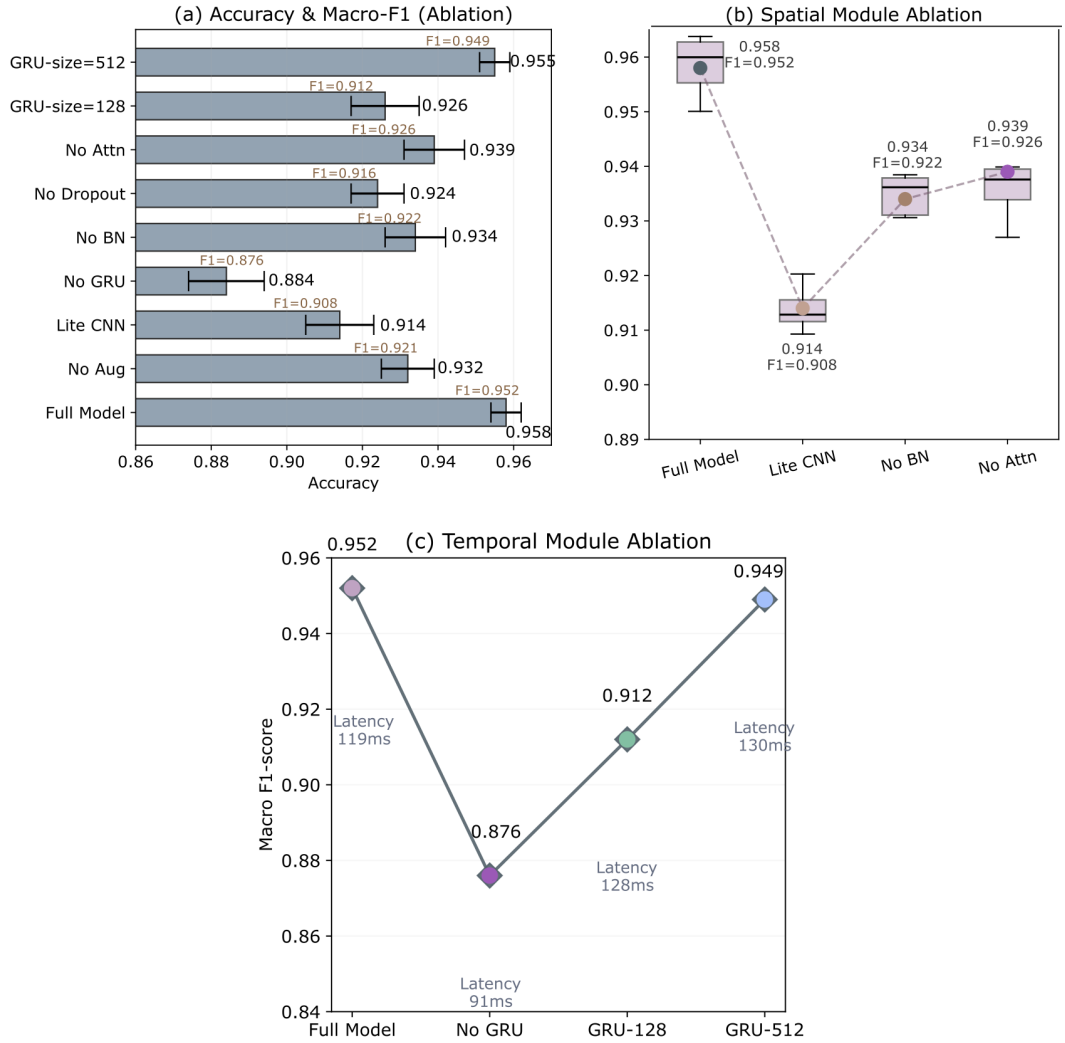

Figure 6. Ablation Study of Model Components(a) Without augmentation(b) VGG16 replaced by lightweight CNN(c) GRU replaced by temporal averaging

Quantitative and Comparative Results

In order to assess the VGG16-GRU model's performance statistically, this article systematically benchmarks a number of common gesture recognition architectures. A comparison framework and selection criteria for baseline approaches have been given in light of the recent development of convolutional, recurrent, and transformer-based models. Multiple times are employed for both training and data sampling to handle random variances, and all results are presented using statistically sound indicators.

The models' overall outcomes are displayed in Figure 7. The VGG16-GRU has a general gesture recognition accuracy of 95.8%, which is little less than the best Vision Transformer model's 96.3%, as seen in Figure 7a. In terms of the macro-averaged F1-score (which represents the performance of both the majority and minority gesture classes), VGG16-GRU achieved a higher value of 95.2% than all convolutional-recurrent baselines and was near the best results obtained by transformers, despite the statistical marginal difference (Figure 7b). As a result, the new structure will be sensitive and precise enough to detect class imbalance in real-world applications.

A multi-dimensional comparison has also been presented, as seen in Figure 7c; the key performance metrics are computational efficiency, macro F1-score, average recognition accuracy, and per-class recall. When compared

to transformer-based models, VGG16-GRU has consistently beaten 3D CNNs and hybrid CNN-LSTM networks in terms of computational resources and energy consumption, making it the most effective non-transformer design overall.

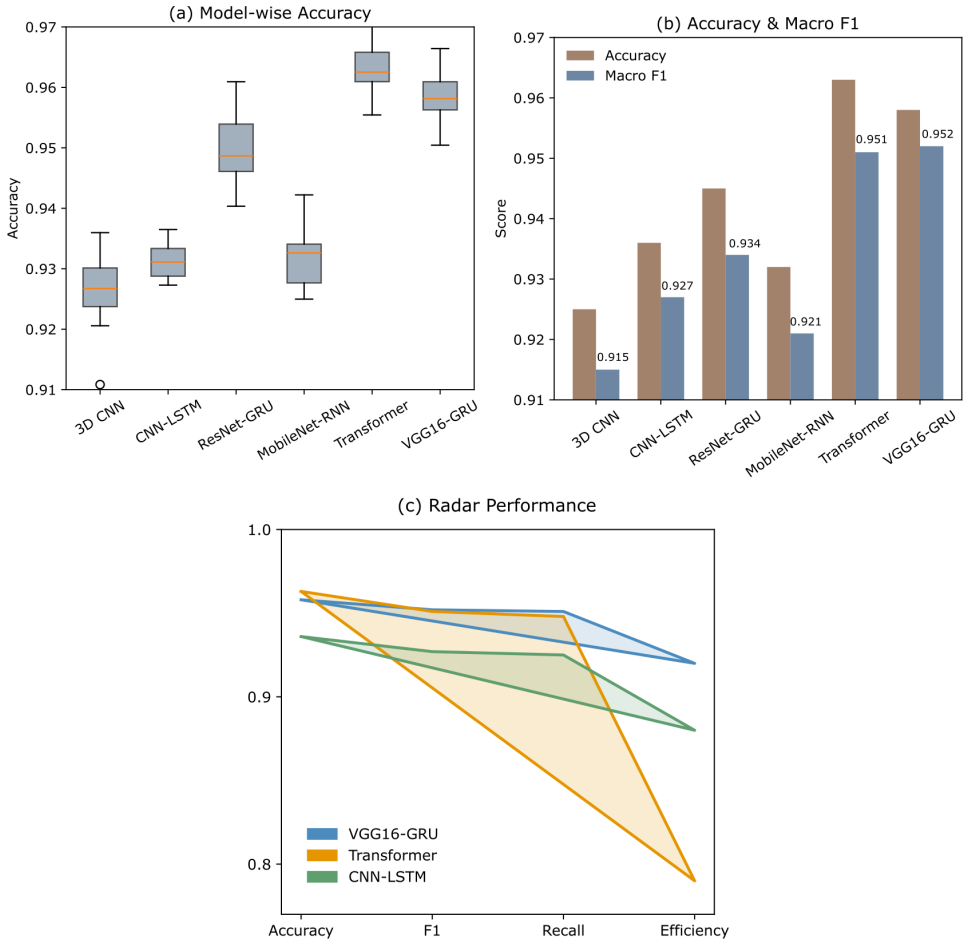

Figure 7. Cross-Architecture Performance Comparison(a) Recognition accuracy comparison(b) Macro F1-score comparison(c) Spider chart of accuracy, F1, recall, and efficiency

They are not just high globally but also rather stable. To evaluate the system's stability under real-time inference constraints and in the presence of noise, additional controlled tests were carried out. The impact of random hand occlusion and artificial motion blur on accuracy is depicted in Figure 8a. Interestingly, as compared to lighter RNN-based models and conventional convolutional backbones, VGG16-GRU has a much lower drop and still performs better than 92% under these perturbations. As a result, people with various movement patterns are likely to benefit from the aforementioned steps.

The system's efficiency breakdown and end-to-end latency versus input sequence length are plotted in Figure 8b. While the latency of transformer models has increased more quickly with longer sequence lengths, the VGG16-GRU model can process up to 40 frames of gesture input in an average of 120 ms, meeting the relatively high real-time needs of most human-computer interaction scenarios.

Although transformer architectures have achieved improved top-line accuracy, their memory and power consumption are relatively substantial when compared to other models when operated on CPUs, as demonstrated in Figure 8c based on an assessment of computational overhead. Conversely, VGG16-GRU is appropriate for edge, mobile, and embedded applications because it strikes a reasonable compromise between recognition performance and deployment practicality.

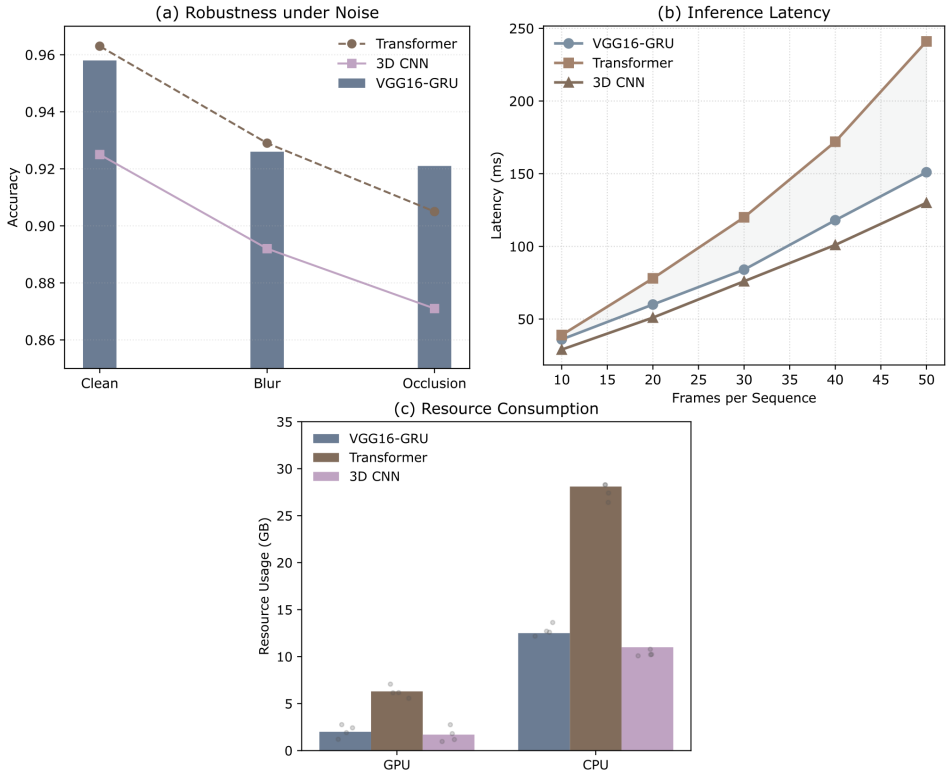

Figure 8. System Robustness and Real-Time Efficiency(a) Accuracy under noise and occlusions(b) Inference latency for varying input lengths(c) Computational resource consumption on GPU and CPU

Conclusion

This study introduces a novel gesture detection system that uses a GRU recurrent layer in conjunction with VGG16 convolutional backbones to merge spatial and temporal models. The VGG16-GRU model has shown exceptional performance, attaining a 95.8% accuracy and one of the highest macro-averaged F1-scores, in an all-around evaluation using both public benchmarks and a real-scene dataset. This demonstrates a significant improvement in addressing contextual fluctuations and class imbalance, both of which have long been recognised issues in gesture recognition studies. The aforementioned empirical findings demonstrate that the system is stable in the face of noise and occlusion, maintains an inference latency of less than 120 ms, and is both accurate in prediction and appropriate for real-time interaction. The aforementioned findings collectively demonstrate that the suggested approach is a sensible option for widespread, useful applications in gesture-driven human-machine interactions.

The study does, however, have the following shortcomings. The well-known temporal drift and memory saturation of recurrent models cause the identification performance to deteriorate for long or unclear gesture sequences that span more than 60 frames. Misclassification of movements with extremely similar looks can still happen when dealing with extreme lighting variations or close-up photos of the hands, even if the additional data augmentation and class-balancing approaches have somewhat decreased the bias. Although there are some efficiency gains over bigger contexts, optimising the model for extremely resource-constrained environments is still an unresolved challenge. Recently, transformer-based designs have been developed [35].

The future work will focus on the following three categories. First, recognition can be preserved in low light by providing a tiny quantity of sensor data, such as a depth image, an inertial sensor, or electromyography signals. Second, in order to increase application on embedded and wearable platforms, research will concentrate on creating a lightweight model using quantisation, pruning, and neural architecture search strategies. Lastly, the use of an enhanced feature attention module can lessen the influence of the background and enhance the concentration on important space-time information. The achievement of these objectives may facilitate the

creation of comprehensive intelligent applications in wearable technology, intelligent industrial settings, and advanced in-car gesture control systems.

Author Contributions

Lucie Hájková contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Li, Q., & Langari, R. (2023). Myoelectric human computer interaction using CNN-LSTM neural network for dynamic hand gesture recognition. Journal of Intelligent & Fuzzy Systems,44(3), 4207-4221. https://doi.org/10.3233/JIFS-222985

[2] Coffen, B., & Mahmud, M. S. (2021, March). Tinydl: Edge computing and deep learning based real-time hand gesture recognition using wearable sensor. In 2020 IEEE international conference on e-health networking, application & services (HEALTHCOM)(pp. 1-6). IEEE. https://doi.org/10.1109/HEALTHCOM49281.2021.9399005

[3] Balaji, P., & Prusty, M. R. (2024). Multimodal fusion hierarchical self-attention network for dynamic hand gesture recognition. Journal of Visual Communication and Image Representation,98, 104019. https://doi.org/10.1016/j.jvcir.2023.104019

[4] Nuzzi, C., Pasinetti, S., Lancini, M., Docchio, F., & Sansoni, G. (2019). Deep learning-based hand gesture recognition for collaborative robots. IEEE Instrumentation & Measurement Magazine,22(2), 44-51. https://doi.org/10.1109/MIM.2019.8674634

[5] Peng, S. H., & Tsai, P. H. (2023). An efficient graph convolution network for skeleton-based dynamic hand gesture recognition. IEEE transactions on cognitive and developmental systems, 15(4), 2179-2189. https://doi.org/10.1109/TCDS.2023.3242988

[6] Wu, W., Shi, M., Wu, T., Zhao, D., Zhang, S., & Li, J. (2019, June). Real-time hand gesture recognition based on deep learning in complex environments. In2019 Chinese Control And Decision Conference (CCDC)(pp. 5950-5955). IEEE. https://doi.org/10.1109/CCDC.2019.8833328

[7] Xing, T., Yang, Q., Jiang, Z., Fu, X., Wang, J., Wu, C. Q., & Chen, X. (2022). WiFine: Real-time gesture recognition using Wi-Fi with edge intelligence. ACM Transactions on Sensor Networks,19(1), 1-24. https://doi.org/10.1145/3532094

[8] Saboo, S., Singha, J., & Laskar, R. H. (2023). Deep learning based spatio-temporal hand gesture recognition system in complex environment.Expert Systems,40(8), e13313. https://doi.org/10.1111/exsy.13313Digital Object Identifier (DOI)

[9] Chakraborty, B. K., Sarma, D., Bhuyan, M. K., & MacDorman, K. F. (2018). Review of constraints on vision-based gesture recognition for human–computer interaction. IET Computer Vision,12(1), 3-15. https://doi.org/10.1049/iet-cvi.2017.0052Digital Object Identifier (DOI)

[10] Köpüklü, O., Gunduz, A., Kose, N., & Rigoll, G. (2020). Online dynamic hand gesture recognition including efficiency analysis.IEEE Transactions on Biometrics, Behavior, and Identity Science,2(2), 85-97. https://doi.org/10.1109/TBIOM.2020.2968216

[11] Ramachandran, S., Shreedar, R. M., & Narayana, K. E. (2024, April). Online Dynamic Hand Gesture Recognition Using 3D-CNN-RNN Hybrid Architecture. In2024 2nd International Conference on Networking and Communications (ICNWC)(pp. 1-6). IEEE. https://doi.org/10.1109/ICNWC60771.2024.10537446

[12] Bao, Z., Zhang, X., Qu, Y., Shao, H., & Qin, G. (2025). An Adaptive Quantization Method for Edge Computing-Based Fault Diagnosis.IEEE Sensors Journal. https://doi.org/10.1109/JSEN.2025.3591151

[13] Gu, Y., Liu, T., Xu, Y., Shen, Y., Ren, H., & Wang, J. (2022, December). A sample data augmentation method for EMG gesture recognition. In2022 2nd International Conference on Electrical Engineering and Control Science (IC2ECS)(pp. 442-446). IEEE. https://doi.org/10.1109/IC2ECS57645.2022.10087946

[14] Hu, F., Qian, M., He, K., Zhang, W. A., & Yang, X. (2024). A novel multi-feature fusion network with spatial partitioning strategy and cross-attention for armband-based gesture recognition. IEEE Transactions on Neural Systems and Rehabilitation Engineering,32, 3878-3890. https://doi.org/10.1109/TNSRE.2024.3487216

[15] Monteiro, S. M., Bota, P., Cunha, P. S., Oliveira, M. M., Laranjo, S., & da Silva, H. P. (2026). Machine learning for the prediction of atrial fibrillation recurrence after catheter ablation: A systematic review and meta-analysis.Computer Methods and Programs in Biomedicine, 109249. https://doi.org/10.1016/j.cmpb.2026.109249

[16] Bhiri, N. M., Ameur, S., Jegham, I., Alouani, I., & Khalifa, A. B. (2024, July). A deep cnn-bigru network for multi-stream hand gesture recognition framework. In2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 893-898). IEEE. https://doi.org/10.1109/CoDIT62066.2024.10708341

[17] Miah, A. S. M., Hasan, M. A. M., & Shin, J. (2023). Dynamic hand gesture recognition using multi-branch attention based graph and general deep learning model. IEEE Access,11, 4703-4716. https://doi.org/10.1109/ACCESS.2023.3235368

[18] Yan, H., Zhang, X., Huang, J., Feng, Y., Li, M., Wang, A., ... & Liu, Z. (2025). Wi-sfdagr: Wifi-based cross-domain gesture recognition via source-free domain adaptation. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2025.3554228

[19] Hax, D. R. T., Penava, P., Krodel, S., Razova, L., & Buettner, R. (2024). A novel hybrid deep learning architecture for dynamic hand gesture recognition. IEEE Access,12, 28761-28774. https://doi.org/10.1109/ACCESS.2024.3365274

[20] Sharma, V., Jaiswal, M., Sharma, A., Saini, S., & Tomar, R. (2021, December). Dynamic two hand gesture recognition using CNN-LSTM based networks. In2021 IEEE international symposium on smart electronic systems (iSES)(pp. 224-229). IEEE. https://doi.org/10.1109/iSES52644.2021.00059

[21] Li, Q., & Langari, R. (2022). EMG-based HCI using CNN-LSTM neural network for dynamic hand gestures recognition. IFAC-PapersOnLine, 55(37), 426-431. https://doi.org/10.1016/j.ifacol.2022.11.220

[22] Malu, G. (2024). Engesto: An ensemble learning approach for classification of hand gestures. IEEE Access, 12, 85709-85723. https://doi.org/10.1109/ACCESS.2024.3411155

[23] Cui, C., Sunar, M. S., & Su, G. E. (2025). Deep vision-based real-time hand gesture recognition: a review. PeerJ Computer Science, 11, e2921. https://doi.org/10.7717/peerj-cs.2921

[24] Liu, D., Wang, T., Liu, S., Wang, R., Yao, S., & Abdelzaher, T. (2021, July). Contrastive self-supervised representation learning for sensing signals from the time-frequency perspective. In 2021 International Conference on Computer Communications and Networks (ICCCN) (pp. 1-10). IEEE. https://doi.org/10.1109/ICCCN52240.2021.9522151

[25] Rahimian, E., Zabihi, S., Asif, A., Farina, D., Atashzar, S. F., & Mohammadi, A. (2022, May). Hand gesture recognition using temporal convolutions and attention mechanism. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1196-1200). IEEE. https://doi.org/10.1109/ICASSP43922.2022.9746174

[26] Esposito, M., Raggiunto, S., Napoletano, P., Belli, A., Sciarroni, M. M., Storti, E., & Pierleoni, P. (2025, September). A Lightweight CNN-Based Solution for Inertial Gesture Recognition on Tiny Edge Devices. In 2025 IEEE 30th International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-7). IEEE. https://doi.org/10.1109/ETFA65518.2025.11205794

[27] Santiago, S. S., & Cifuentes, J. A. (2024). Deep learning-based gesture recognition for surgical applications: A data augmentation approach. Expert Systems,41(12), e13706. https://doi.org/10.1111/exsy.13706

[28] Gammulle, H., Denman, S., Sridharan, S., & Fookes, C. (2021). TMMF: Temporal multi-modal fusion for single-stage continuous gesture recognition. IEEE Transactions on Image Processing, 30, 7689-7701. https://doi.org/10.1109/TIP.2021.3108349

[29] Köpüklü, O., Ledwon, T., Rong, Y., Kose, N., & Rigoll, G. (2020, November). Drivermhg: A multi-modal dataset for dynamic recognition of driver micro hand gestures and a real-time recognition framework. In 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020) (pp. 77-84). IEEE. https://doi.org/10.1109/FG47880.2020.00041

[30] Fallahhusein, M., Nandy, M., Kumar, S. S., Kumar, C. V., Bhanu, D., Al-Dulaimi, H. W., ... & Yehya, M. (2025, July). Real-Time Gesture Recognition Algorithm Using CNN and LSTM for Secure Human-Computer Interaction. In 2025 3rd International Conference on Cyber Resilience (ICCR) (pp. 1-7). IEEE. https://doi.org/10.1109/ICCR67387.2025.11291976