

Article

Gesture Classification Using a Smartwatch: Focusing on Unseen Non-Target Gestures

Jae-Hyuk Choi, Hyun-Tae Choi, Kyeong-Taek Kim [†], Jin-Sub Jung, Seok-Hyeon Lee and Won-Du Chang ^{*ID}

Department of Computer Engineering, Pukyong National University, Busan 48513, Republic of Korea; tjzjs12@pukyong.ac.kr (J.-H.C.); chlgus1233@pukyong.ac.kr (H.-T.C.); kkt0098@gmail.com (K.-T.K.); piandphi@naver.com (J.-S.J.); ha2sted@gmail.com (S.-H.L.)

^{*} Correspondence: chang@pknu.ac.kr

[†] Current address: Nexen Tire Research Center, 177 Magoh-ro, Gangseo-gu, Seoul 07594, Republic of Korea.

Abstract: Hand gestures serve as a fundamental means of communication, and extensive research has been conducted to develop automated recognition systems. These systems are expected to improve human/computer interaction, particularly in environments where verbal communication is limited. A key challenge in these systems is the classification of non-target actions, as everyday movements are often not included in the training set, but resembling target gestures can lead to misclassification. Unlike previous studies that primarily focused on target action recognition, this study explicitly addresses the unseen non-target classification problem through an experiment to distinguish target and non-target activities based on movement characteristics. This study examines the ability of deep learning models to generalize classification criteria beyond predefined training sets. The proposed method was validated with arm movement data from 20 target group participants and 11 non-target group participants, achieving an average F1-score of 84.23%, with a non-target classification score of 73.23%. Furthermore, we confirmed that data augmentation and incorporating a loss factor significantly improved the recognition of unseen non-target gestures. The results suggest that improving classification performance on untrained, non-target movements will enhance the applicability of gesture recognition systems in real-world environments. This is particularly relevant for wearable devices, assistive technologies, and human/computer interaction systems.



Academic Editor: Giuseppe Andreoni

Received: 3 February 2025

Revised: 2 April 2025

Accepted: 24 April 2025

Published: 27 April 2025

Citation: Choi, J.-H.; Choi, H.-T.; Kim, K.-T.; Jung, J.-S.; Lee, S.-H.; Chang, W.-D. Gesture Classification Using a Smartwatch: Focusing on Unseen Non-Target Gestures. *Appl. Sci.* **2025**, *15*, 4867. <https://doi.org/10.3390/app15094867>

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/>).

Keywords: pattern recognition; deep neural network; gesture recognition; human/computer interface; zero-shot approach

1. Introduction

Hand gestures are a significant aspect of human communication, enriching interactions by offering additional expressive capabilities. They allow individuals to visually convey emotions or intentions that may be challenging to express through speech. In certain contexts, gestures even serve as an effective alternative to verbal communication or complement it. They are particularly useful in environments with high noise levels, where visual communication is more practical, and in settings that require silence. Furthermore, hand gestures provide a valuable means of communication for individuals with speech impairments or pronunciation difficulties.

In recent years, hand gestures have been increasingly explored as a means of interaction between humans and computers (HCI). Automatic recognition of hand gestures enables the recording, analysis, and classification of specific hand or arm movements using computer algorithms. The most prevalent methodology for capturing hand gestures is the

camera-based approach, which employs an optical camera. This approach aims to detect entire hands, fingertips, pens, or handheld devices.

Hand gestures for HCI are often designed without finger or wrist movements when the number of commands is limited. In such cases, the accuracy of hand gesture recognition systems in camera-based approaches primarily depends on the precise detection of pen tips or fingertips. Once the tips are accurately detected, hand positions can be easily tracked across consecutive video frames [1]. This tracking enables hand gesture recognition to be treated similarly to handwritten character recognition, which has achieved high levels of accuracy over the past few decades. Various techniques have been proposed, including linear discriminant analysis, multi-layer perceptron, tangent distance, and support vector machines [2,3]. Recent research indicates that convolutional neural networks can achieve accuracies exceeding 99% in handwritten character recognition [4,5].

Consequently, various techniques have been explored to enhance hand detection accuracies in camera-based systems. Color information was effectively utilized to identify the pen tip position in [6], assuming a fixed pen color during the experiments. A similar approach was employed to detect hand regions by predefining hand color in a controlled environment [7].

To further improve hand gesture recognition, the detection of hands or fingertips can be enhanced by replacing the optical camera with a depth sensor, which provides additional depth information about the object. For instance, the software development kit (SDK) of a Kinect sensor was utilized to detect hand positions, while the fingertip positions were calculated from hand and depth data [8,9]. The Kinect sensor has also been used to track fingertip positions and recognize tapping gestures, demonstrating its effectiveness in gesture recognition [10]. Similarly, an alternative approach to hand tracking involving depth measurement with an ultrasonic transceiver has been introduced [11].

Despite these advancements, camera- and depth sensor-based methods have inherent limitations, as they typically require users to remain within the field of view of the sensor. Although head-mounted cameras, as proposed in [12], alleviate this constraint to some extent, they still present practical challenges.

An alternative method for recording hand gestures involves the use of accelerometer and gyroscope sensors, which provide users with greater freedom of movement. As these sensors are integrated into most modern smartwatches and fitness bands, users can utilize them as input devices with minimal effort. Furthermore, this approach remains stable regardless of lighting conditions.

However, a major challenge associated with accelerometer- and gyroscope-based methods is the accurate determination of sensor positions within the global coordinate system. Due to sensor movement or rotation during data acquisition, accelerometer values are aligned with the local coordinate system of the sensors. To address this issue, an automated coordinate conversion methodology was proposed in [12]. However, this approach requires the availability of ground truth traces, which are often not readily accessible.

The accuracy of accelerometer- and gyroscope-based methods is inferior to that of camera-based methods. For instance, camera-based approaches have achieved a maximum accuracy of 99.571% for Arabic numbers and 16 directional symbols [7] and 95% for English letters and Arabic numbers [8]. In particular, 98.1% accuracy was achieved for six directive gestures at distances of up to 25 m. This result was obtained using a Graph Vision Transformer (GViT) with an RGB camera and HQ-Net for super-resolution [13]. Additionally, an accuracy of 77% was recorded for 15 sign language gestures using SNN with STBP on an event-camera dataset (DVS_Sign_v2e) [14]. In contrast, accelerometer- and gyroscope-based methods achieved 79.6% accuracy for Japanese Katakana (46 letters) using the K-nearest neighbor (K-NN) method [15], 88.1% for Japanese Hiragana (46 letters)

with the hidden Markov model (HMM) [16], and 89.2% by combining K-NN with dynamic time warping (DTW) techniques [17]. Recently, sensor-based approaches have been improved through deep learning models, such as ABSDA-CNN with a 64-channel EMG sensor, achieving an accuracy of 88.91% for inter-session gesture recognition [18]. Some sensor-based approaches have achieved high accuracy by integrating multiple devices. For instance, Shin et al. achieved an accuracy of 95.3% using a combination of RGB cameras, depth sensors, EMG, and EEG [19].

Sensor-based gesture recognition systems are often expected to operate asynchronously without the need for explicit triggers. In such cases, distinguishing between target and non-target classes is crucial to prevent unintended activations and ensure accurate recognition. Specifically, non-target classes consist of actions that resemble target gestures but commonly occur in everyday activities, making them more challenging to differentiate. Proper identification of these patterns is essential to minimize false positives and ensure reliable system performance in real-world scenarios. Previous research has predominantly ignored non-target actions, collecting and training data exclusively for target actions. However, it is also important to prevent non-target actions from being classified as target actions in addition to recognizing target actions.

Classifying non-target actions can be considered a form of out-of-distribution (OOD) detection or anomaly detection, which has been widely studied in recent decades. Table 1 summarizes representative works [20–32] in these domains. Much of the prior research has focused on image-based tasks [20–22,26,30], while studies on time-series data, such as sensors and gestures, have been relatively limited. These studies on time-series signals have primarily addressed anomaly detection by identifying irregularities within temporal patterns [20,23–25,31]. In addition, some efforts have been made to improve the robustness of classification models to abnormal variations [28,29]. Recent findings further suggest that conventional OOD detection approaches are insufficient for handling complex time-series scenarios involving unseen semantic categories [32]. Zhang et al. addressed this by treating certain time-series classes as OOD and evaluating the model’s ability to recognize them [29]. In their study, a subset of classes within a multi-class dataset was designated as OOD, and the model was tested on its ability to classify these unseen OOD classes.

Table 1. Related studies on out-of-distribution (OOD) detection.

No.	Year	Data Type	Main Contribution	Detection Type	Nature of Abnormal Data
[20]	2019	Mixed (images, time-series, text)	Survey on deep learning for general anomaly detection	Anomaly detection	Anomalies within data or sequence
[21]	2020	Images (CIFAR, TinyImageNet)	OOD detection without using any OOD data via confidence-based methods	OOD classification	Unseen semantic categories
[22]	2021	Images (ImageNet, SVHN)	Evaluation and analysis of the limits of OOD detection methods	OOD classification	Unseen semantic categories
[23]	2021	Time-series (sensors, finance)	Survey of DL techniques for time-series anomaly detection	Anomaly detection	Anomalies within sequences
[24]	2021	Time-series (sensors)	Survey of anomaly detection in multivariate time-series	Anomaly detection	Anomalies within sequences
[25]	2023	Time-series (ECG, UCR, etc.)	Survey of deep learning approaches for time-series anomaly detection	Anomaly detection	Anomalies within sequences

Table 1. Cont.

No.	Year	Data Type	Main Contribution	Detection Type	Nature of Abnormal Data
[26]	2024	Medical images (CT, MRI)	Survey of OOD detection techniques in medical image analysis	OOD classification	Unseen semantic categories
[27]	2024	Time-series (particle traces)	Reliable deep learning under OOD dynamics in anomalous diffusion systems	Anomaly detection	Anomalies within sequences
[28]	2024	Time-series (health records)	Invariant learning for generalization in time-series forecasting	Non-OOD classification	Distributional shift across environments
[29]	2024	Time-series (gesture, speech, wearable signals)	Framework for OOD detection and robust classification under distributional shift	OOD detection and non-OOD classification	Unseen semantic categories
[30]	2024	Image (CIFAR-10, CIFAR-100, SVHN, etc.)	Ensemble-based OOD detection	OOD classification	Unseen semantic categories
[31]	2024	Time-series (ECG, Power Demand, Space Shuttle, etc.)	Contrastive learning with distributional augmentation	Anomaly detection	Anomalies within sequences
[32]	2025	Time-series (epilepsy, handwriting, etc.)	Survey of OOD methods for time-series signals	OOD classification	Unseen semantic categories

Building on this, the present study explores a more challenging setting in which OOD classes exhibit patterns that closely resemble those of the target classes, thereby increasing the difficulty of discrimination. OOD classes were designed to have similar patterns to the target classes and were tested while they were untrained. The proposed approach focuses on evaluating various training strategies to achieve reliable recognition of non-target activities.

The remainder of this paper is organized as follows. The collected data and proposed method are described in Section 2, followed by the presentation of experimental results and related discussions in Section 3. Finally, the study concludes with Section 4.

2. Materials and Methods

This section describes the dataset, preprocessing procedures, neural network architecture, and experimental design employed in this study.

2.1. Dataset

In this study, data were collected from 20 participants aged between 19 and 27. All participants were right-handed and wore a smartwatch on their right wrist during the experiment. The study protocol was approved by the Institutional Review Board (IRB) of Pukyong National University (#1041386-202104-HR-12-01).

The collected data captured right-arm movements in a three-dimensional space, encompassing motion characteristics in the anterior/posterior, medial/lateral, and vertical directions. The target movements were performed according to predefined directional instructions, which included upper-left, upper-right, lower-left, lower-right, and triangular patterns. Each movement consisted of three distinct sub-movements, with a brief pause of approximately 0.5 s between sub-movements to emphasize its distinctive features. All target patterns incorporated pauses and directional changes at either right or acute angles.

The movements were labeled using alphabetical notations, where “F” denotes a forward movement, “R” and “L” indicate rightward and leftward movements, and “U” and “D” represent upward and downward movements. The target movement set included four fundamental patterns (FRU, FRD, FLU, FLD) along with a triangular movement pattern (TRA) (Figure 1a). The forward movement was added to the four fundamental patterns to serve as a potential trigger for future studies. This design also ensures that each pattern consists of three segments, making them structurally comparable to the triangular movement (TRA). Each participant performed 5 distinct target movements 10 times, resulting in a total of 1000 target movement samples.

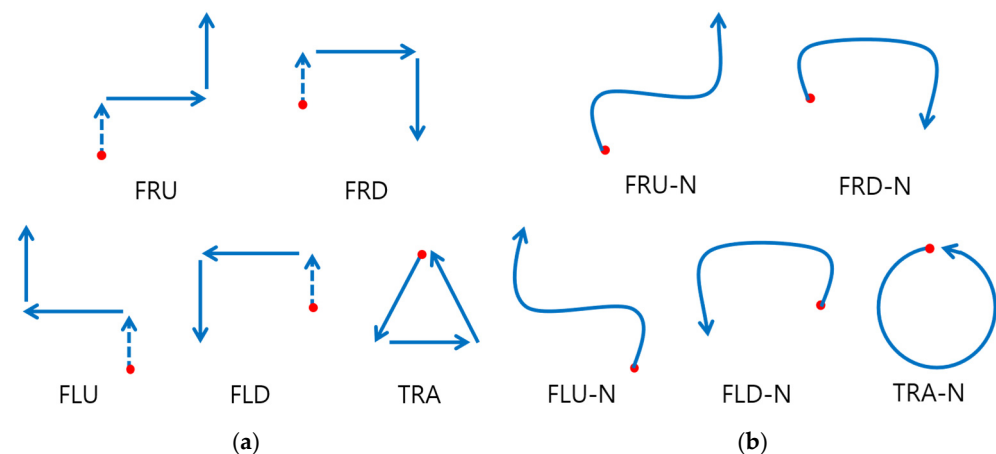


Figure 1. Gesture patterns for targets and non-targets. (a) Target motion movements, where dashed arrows indicate forward movements. (b) Non-target motion movements with curved trajectories. Red dots indicate the starting points of each movement.

Non-target movements are defined as actions that do not correspond to the predefined target movements. These movements closely resemble target movements, which increases the likelihood of misclassification. Non-target movements were derived from the five target movements by intentionally omitting specific distinguishing features, such as pauses or directional changes. Each non-target movement corresponds to a specific target movement.

The primary objective of designing non-target movements was to assess whether the model could accurately classify unseen non-target movements by understanding the general characteristics of target and non-target movements. Unlike target movements, which include momentary pauses at corners and sharp directional changes, non-target movements were executed in a continuous and smooth, curved manner (Figure 1b). For instance, in the FRU, FRD, FLU, FLD, and TRA movements, the non-target versions were designed to move fluidly without interruptions, with the triangular pattern replaced by a circular motion.

Non-target movement data were collected from 11 out of the 20 participants, with each participant performing 15 non-target movements, yielding a total of 165 non-target samples.

The data were collected using sensors embedded in a smartwatch, which measured accelerometer and gyroscope values in three-dimensional space. Each movement was recorded for a duration of three seconds at a sampling rate of 50 Hz. The collected data were transmitted from the smartwatch to a smartphone and subsequently transferred to a PC via a wired connection or Wi-Fi. The sampling rate was limited to 50 Hz due to transmission constraints between the smartwatch and the smartphone used in this experiment, which occasionally caused data loss at higher frequencies.

The experimental setup consisted of an LG Watch W7 smartwatch (LG Electronics, Seoul, Republic of Korea), a Nexus 5X smartphone (LG Electronics, Seoul, Republic of Korea), and a PC. Data collection was conducted using Android Studio Arctic Fox (Figure 2).

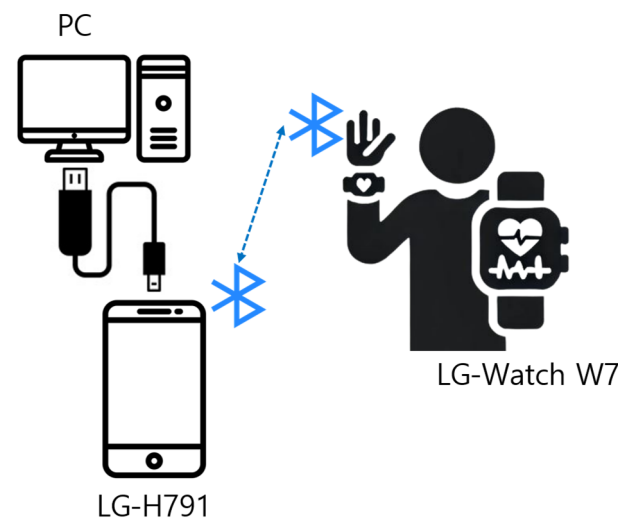


Figure 2. Data communication flow. Sensor signals are transmitted to a mobile phone via Bluetooth and recorded on a personal computer connected by USB.

2.2. Data Standardization and Data Augmentation

The collected data varied in length due to measurement errors from the recording device. To ensure consistency, all samples were standardized to a fixed length of 150 using zero-padding. If a sample had fewer than 150 values, the existing data remained unchanged, and the missing values were filled with zeros to reach 150. Zero-padding was applied to enable consistent input dimensions for convolutional layers, while dropout layers were used to mitigate potential overfitting or artifacts caused by the added padding signals.

Data augmentation is a widely used technique in machine learning to artificially expand the training dataset by applying various transformations to the existing data [33–35]. This approach enhances model generalization by introducing variability and reducing overfitting, which is particularly beneficial in time-series data, where data collection can be costly and time-consuming. Several augmentation techniques—such as Gaussian Noise [33], Cut-out [34], Time Shift [35], and Window Warp [36]—have been utilized for time-series data to enhance its diversity. In the context of this study, data augmentation is expected to improve the model’s robustness in recognizing both target and non-target movements under varying conditions.

The dataset consists of six classes; five classes correspond to the predefined target movements, while the sixth class includes all non-target movements. This classification enables the model to distinguish specific gestures from a broad range of non-target movements.

2.3. Neural Network Architecture

A deep learning model is proposed for the classification of unseen non-target movements, as illustrated in Figure 3. The proposed model is based on a combination of convolutional neural networks (CNN) and bidirectional long short-term memory (Bi-LSTM) layers, which are well-suited for analyzing time-series data. The network architecture comprises four convolutional blocks, where each block consists of three one-dimensional (1D) convolutional layers followed by a max-pooling layer. The convolutional layers in each block have a kernel size of 3, with the number of filters progressively increasing to 8, 16, 32, and 64 across the four blocks. Each convolutional block is followed by a max-pooling

layer that applies a pooling size of 3 with a stride of 2, which progressively reduces the dimensionality of the feature maps.

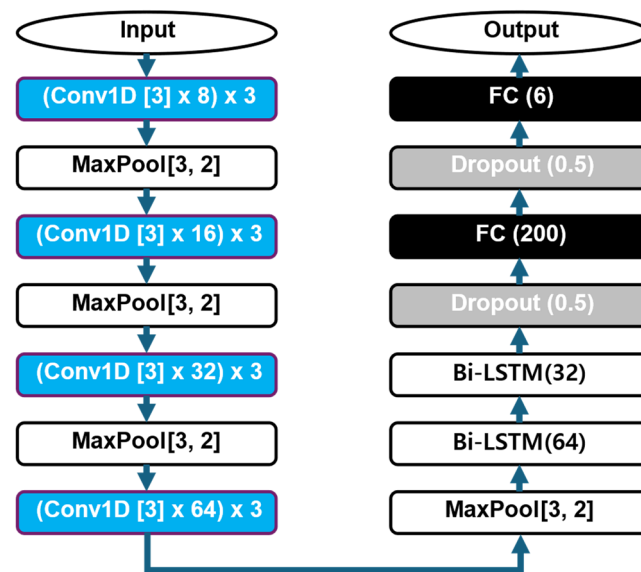


Figure 3. Network architecture for gesture classification. Blue arrows show data flow; blue blocks are convolutional layers, and gray blocks are dropout layers.

After the final convolutional block, the model incorporates two bidirectional LSTM layers with 64 and 32 units, respectively, to effectively capture temporal dependencies in the input sequences. Subsequently, two dropout layers are applied with a dropout ratio of 0.5 to prevent overfitting during training. The final classification is performed through two fully connected layers. The first layer contains 200 units and the output layer consists of 6 units, corresponding to the classification of target and non-target movements. The filter sizes, number of filters, and dropout ratios were empirically determined through extensive experimental evaluations to optimize the model's performance.

The choice of this architecture was guided by the need to balance classification performance and computational efficiency, particularly given the size and characteristics of the dataset. The CNN module captures local spatial features of the movement signals, while the Bi-LSTM layers learn temporal dependencies by utilizing both past and future signals. This combination enhances the model's ability to distinguish between target and non-target movements. A combination of CNN and Bi-LSTM was chosen over more recent architectures such as Transformer or GRU because it provides a good trade-off between model complexity and accuracy for relatively small-scale gesture data. In preliminary trials, the proposed architecture showed stable training and competitive accuracy, making it suitable for the problem setting in this study.

The proposed model utilizes a four-layer 1D CNN with increasing filter sizes (8, 16, 32, 64) to extract spatial features, followed by max-pooling (3, 2) and dropout (0.5) to reduce overfitting. These architectural choices were based on prior studies in time-series classification and confirmed through preliminary experiments. A kernel size of 3 was selected to capture fine-grained local features, while the increasing number of filters across layers allows the model to learn progressively higher-level representations. The dropout rate of 0.5 is a widely used heuristic that provides a good balance between regularization and model capacity. Bidirectional LSTM layers (64, 32 units) capture temporal dependencies in both directions, and fully connected layers (200, 6) perform final classification, maintaining a balance between efficiency and generalization.

A loss function was designed to improve classification performance by considering both class identification errors and the distinction between target and non-target patterns. The total loss function consists of two components: the categorical cross-entropy loss for class identification and an additional loss term that enhances the separation between target and non-target classes.

The first component, the class identification loss, is formulated as follows:

$$L_{class} = -\frac{1}{N \cdot C} \sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log \hat{y}_{i,c}, \quad (1)$$

where N represents the total number of samples, C denotes the number of classes. $y_{i,c}$ is the ground-truth label for sample i and class c , and $\hat{y}_{i,c}$ is the predicted probability of the sample belonging to class c . This term ensures that the model effectively learns to classify each sample into the correct class by minimizing the negative log-likelihood of the correct class probability.

To reinforce the distinction between target and non-target patterns, an additional target loss term is introduced. This term is defined as:

$$L_{target} = -\frac{1}{2N} \sum_{i=1}^N (t_{i,1} \log \hat{t}_{i,1} + t_{i,2} \log \hat{t}_{i,2}), \quad (2)$$

where $t_{i,1}$ and $t_{i,2}$ represent the target labels, indicating whether the sample belongs to one of the five target classes (FRU, FRD, FLU, FLD, and TRA) or the non-target class. All the non-target classes are considered as a single class. The predicted probability values $\hat{t}_{i,1}$ and $\hat{t}_{i,2}$ are obtained by aggregating the softmax outputs for the target classes into a single probability while maintaining the probability of the non-target class. These predicted target values, $\hat{t}_{i,1}$ and $\hat{t}_{i,2}$, are defined as

$$\hat{t}_{i,1} = \sum_{k=1}^5 p_{i,k}, \quad (3)$$

$$\hat{t}_{i,2} = p_{i,6}, \quad (4)$$

where $p_{i,c}$ denotes the predicted output of the model for each class. This formulation is designed to ensure that the model explicitly learns to distinguish between target and non-target patterns.

The final loss function is expressed as the weighted sum of these two terms:

$$L_{total} = L_{class} + \lambda \cdot L_{target}, \quad (5)$$

where λ is a weighting factor that controls the contribution of the target loss term to the overall loss. By incorporating both loss terms, the proposed loss function aims to enhance classification accuracy while ensuring a robust differentiation between target and non-target classes.

2.4. Experimental Setting

In this paper, we aim to assess the model's ability to distinguish between target and non-target movements effectively. To achieve this, the model was trained using only four classes of non-target movements. The remaining class was excluded from the training phase to evaluate its ability to recognize unseen non-target movements (Figure 4). The experimental dataset consists of five target movements and five non-target movements, which were utilized to train and evaluate the model. During the testing phase, the trained model was evaluated with unseen non-target movements to determine how well they were classified.

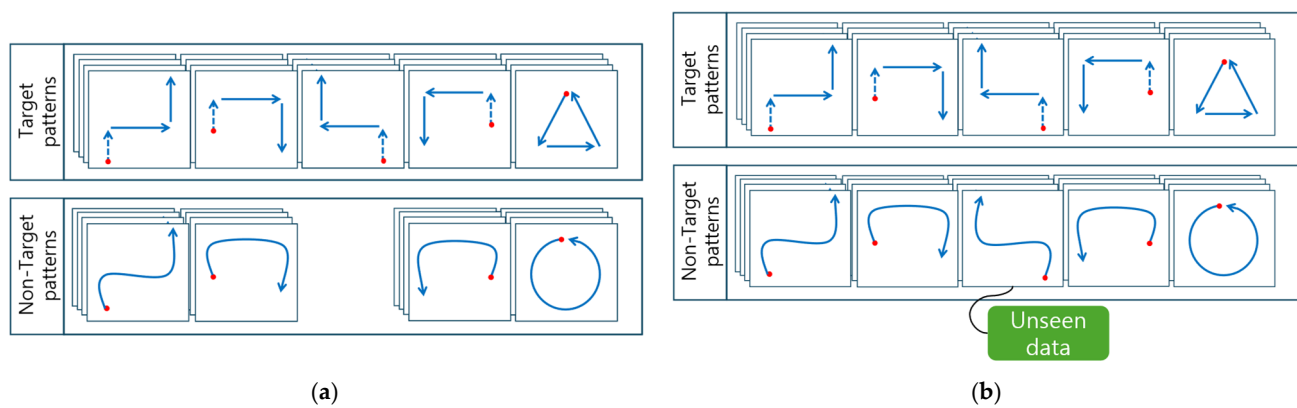


Figure 4. Concept of validating unseen non-target data. A type of non-target pattern is excluded from the training dataset, while all patterns are evaluated during testing. (a) Training phase. (b) Testing phase. Red dots indicate the starting points of each movement.

Each non-target class consisted of a total of 33 samples, resulting in 132 samples being used for training and the remaining 33 samples for testing (Table 2). A similar approach was applied to the target movements, where 33 samples were allocated for testing, and the remaining samples were used for training. To ensure the reliability of the validation process, the selection of target data was performed randomly and repeated 10 times. For non-target patterns, the experiment was repeated five times by leaving each class as an unseen class in turn. Consequently, a total of 50 experimental iterations were conducted for each model, enhancing the robustness of the evaluation.

Table 2. Repeated training and testing splits for target and non-target classes.

Trial	Training										Test									
	Target Classes					Non-Target Classes					Targets					Non-Target Classes				
	FRU	FRD	FLU	FLD	TRA	FRU	FRD	FLU	FLD	TRA	FRU	FRD	FLU	FLD	TRA	FRU	FRD	FLU	FLD	TRA
1	167 for each class					0	132	132	132	132	33 for each class					33	33	33	33	33
2						132	0	132	132	132						33	33	33	33	33
3						132	132	0	132	132						33	33	33	33	33
4						132	132	132	0	132						33	33	33	33	33
5						132	132	132	132	0						33	33	33	33	33

3. Results

The classification performance of the proposed model was evaluated using precision, recall, and F1-score metrics (Table 3). The results indicate that the model achieved an overall precision, recall, and F1-score of 0.8592, 0.8459, and 0.8423, respectively. When considering only target patterns, the model exhibited an average precision of 0.8497, recall of 0.8885, and F1-score of 0.8643. In contrast, for non-target patterns, the model achieved a higher precision of 0.9071 but significantly lower recall (0.6327) and F1-score (0.7323). This discrepancy suggests that while the model is effective in maintaining high precision for non-target patterns, it tends to exhibit lower recall, implying that some non-target patterns are misclassified as target patterns.

Table 3. Model performance on target and unseen non-target patterns.

Training		Test	Accuracy (Averaged over 10 Repetition)		
Target Patterns	Non-Target Patterns	Non-Target Patterns	Precision	Recall	F1 Score
All target patterns	All except FRU-N	FRU	0.7698	0.8636	0.8107
		FRD	0.8487	0.8879	0.8663
		FLU	0.8977	0.7879	0.8379
		FLD	0.8161	0.8939	0.8523
		TRA	0.9732	0.9727	0.9728
		FRU-N	0.9365	0.7758	0.8411
		Average	0.8737	0.8636	0.8635
	All except FRD-N	FRU	0.8801	0.8697	0.8743
		FRD	0.7729	0.9030	0.8310
		FLU	0.8828	0.8121	0.8444
		FLD	0.8237	0.9212	0.8688
		TRA	0.9208	0.9667	0.9422
		FRD-N	0.9111	0.6697	0.7663
		Average	0.8652	0.8571	0.8545
	All except FLU-N	FRU	0.9107	0.8364	0.8696
		FRD	0.8366	0.8727	0.8514
		FLU	0.7595	0.8879	0.8163
		FLD	0.8016	0.9000	0.8452
		TRA	0.9494	0.9667	0.9566
		FLU-N	0.9173	0.6303	0.7385
		Average	0.8625	0.8490	0.8463
	All except FLD-N	FRU	0.8509	0.8424	0.8453
		FRD	0.8452	0.8697	0.8563
		FLU	0.8524	0.8152	0.8318
		FLD	0.7638	0.9152	0.8323
		TRA	0.9253	0.9606	0.9422
		FLD-N	0.9242	0.7121	0.8016
		Average	0.8603	0.8525	0.8516
	All except TRA-N	FRU	0.8880	0.8848	0.8862
		FRD	0.8822	0.8848	0.8805
		FLU	0.9036	0.7970	0.8453
		FLD	0.8686	0.9061	0.8855
		TRA	0.6178	0.9939	0.7612
		TRA-N	0.8462	0.3758	0.5141
		Average	0.8344	0.8071	0.7955
Average (total)			0.8592	0.8459	0.8423
Average (targets)			0.8497	0.8885	0.8643
Average (non-targets)			0.9071	0.6327	0.7323

A detailed analysis of individual test conditions reveals variations in model performance across different excluded non-target patterns. The model's recall is notably lower when classifying the TRA-N pattern, with a value of 0.3758, resulting in an F1-score of 0.5141. This phenomenon appears to be attributed to the unique characteristics of the TRA pattern. Unlike other movement patterns, which follow distinct directional shifts, the TRA pattern is designed with a continuous circular motion. Consequently, when the TRA-N pattern is excluded from the training data, the model struggles to classify it accurately during testing. In contrast, other non-target patterns maintain a similar fundamental movement style, differing only in direction, making their classification less challenging.

Among target patterns, the model performed consistently well, achieving the highest F1-score of 0.9728 for the TRA movement, while the lowest F1-score was observed for the FLU movement (0.8379). The model's performance on unseen non-target patterns also varied depending on the excluded class, with the highest F1-score (0.8016) observed for the FLD-N pattern and the lowest (0.5141) for TRA-N.

To further evaluate the classification performance of the proposed model, confusion matrices were analyzed across the five different experimental settings. Each setting was validated ten times (Figure 5). The rows indicate the actual class labels, while the columns represent the predicted class labels.

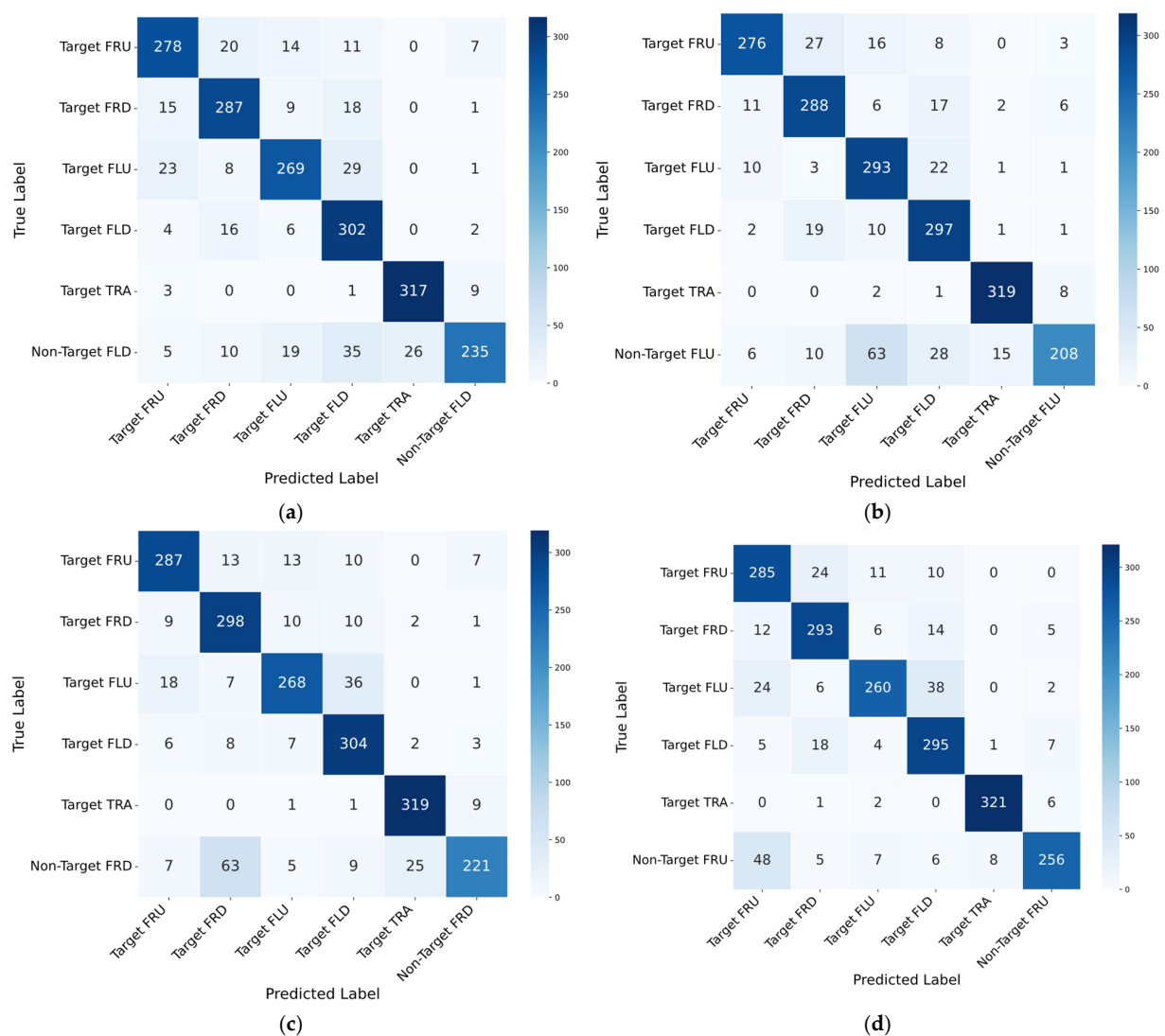


Figure 5. Cont.

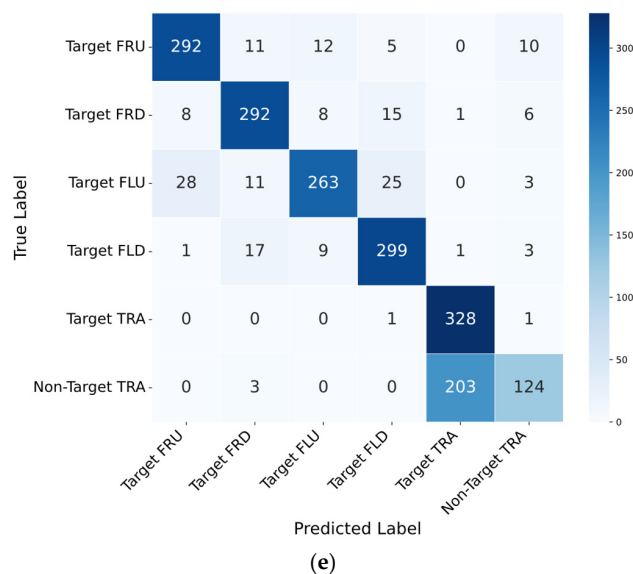


Figure 5. Confusion matrices of the proposed method with different unseen patterns: (a) Non-Target FRU, (b) Non-Target FRD, (c) Non-Target FLU, (d) Non-Target FLD, (e) Non-Target TRA.

The overall results indicate that the model effectively classifies target movements, with most instances correctly assigned to their respective target categories. However, variations in misclassification patterns were observed depending on the excluded non-target pattern. In particular, the model demonstrated strong classification performance for the TRA movement, consistently achieving high accuracy across all settings. The distinct motion characteristics of this pattern likely contribute to its robust classification. Conversely, FLU and FRU exhibited relatively higher misclassification rates, primarily due to their similar movement trajectories. This pattern of confusion suggests that the model struggles to distinguish between certain target movements when they exhibit overlapping motion characteristics.

Figure 6 presents the accuracy and loss curves for both the training and validation phases. Although an increase in validation loss was observed, this does not necessarily indicate deteriorating performance. The rise in loss could occur due to correct predictions made with low confidence. The smooth convergence of the accuracy curves supports the model's stability throughout training.

Table 4 presents the classification performance of the proposed model with different data augmentation methods. The results indicate that data augmentation positively influenced model performance, with varying degrees of improvement depending on the augmentation technique used. Without augmentation, the model achieved an average F1-score of 0.6604, with relatively poor performance on non-target patterns. This suggests that the model struggles to generalize effectively to unseen non-target movements in the absence of augmentation.

Among the tested augmentation methods, Gaussian Noise led to the highest overall performance, with an average F1-score of 0.7323. This method significantly improved classification accuracy across all non-target patterns, particularly FRU-N (0.8411) and FLD-N (0.8016). The improved performance suggests that introducing noise enhances the model's robustness by preventing overfitting to specific movement characteristics.

Time Shift augmentation also improved model performance, achieving an average F1-score of 0.6897. Notably, this method enhanced the classification of FRD-N (0.8130) and TRA-N (0.6439), suggesting that shifting temporal features allows the model to better capture movement variations. However, performance on FLU-N (0.5850) and FLD-N

(0.6172) remained relatively low, indicating that some patterns may be more sensitive to temporal shifts.

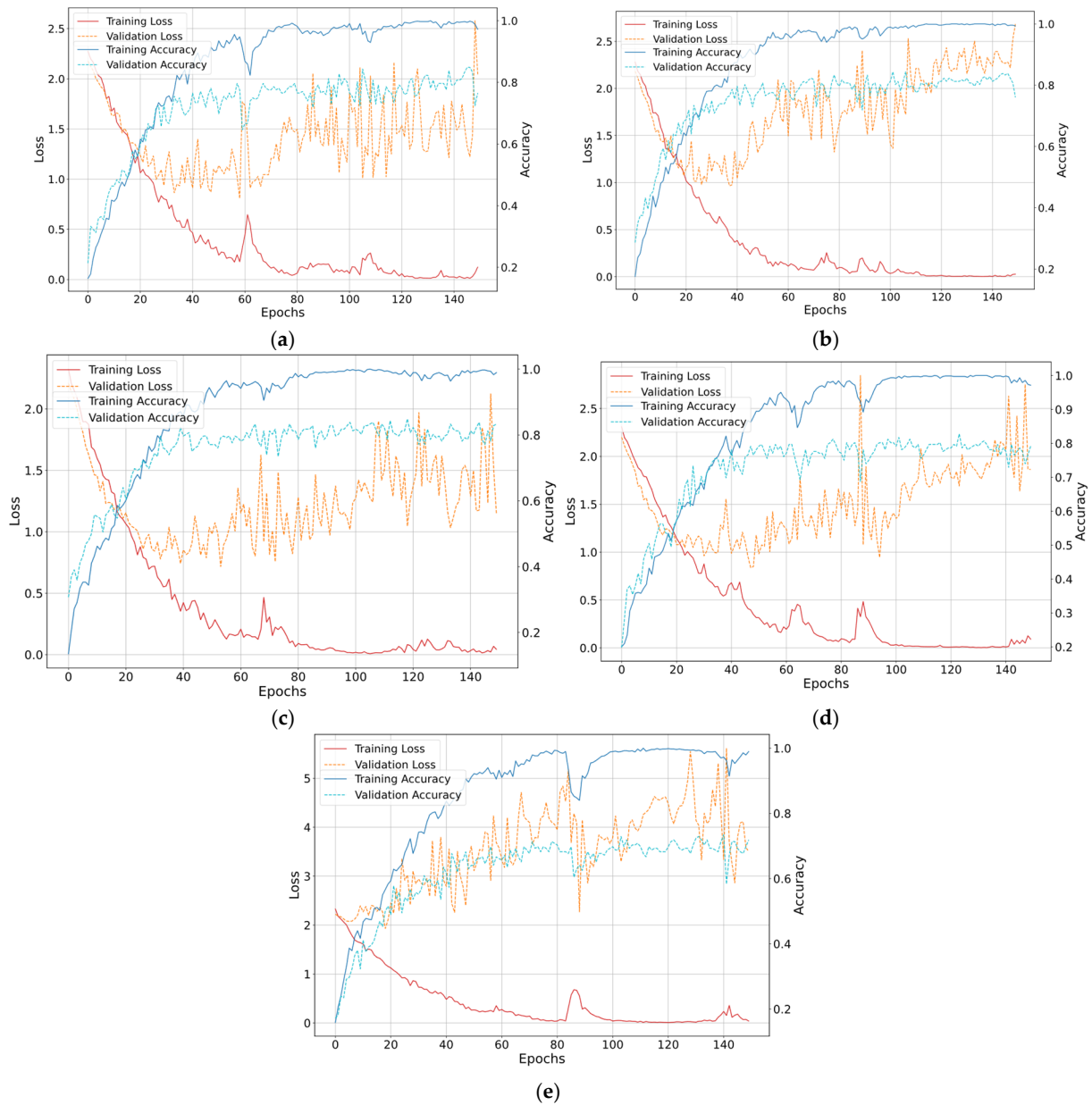


Figure 6. Loss and accuracy curves for training and validation on different unseen patterns: (a) Non-Target FLD, (b) Non-Target FLU, (c) Non-Target FRD, (d) Non-Target FRU, (e) Non-Target TRA.

Table 4. Model performance on unseen non-target patterns with different augmentation methods.

	FRU-N	FRD-N	FLU-N	FLD-N	TRA-N	Mean $\pm$ Std
No augmentation	0.7732 ± 0.0668	0.6292 ± 0.1211	0.6878 ± 0.0676	0.7103 ± 0.0916	0.5015 ± 0.0795	0.6604 ± 0.1252
Time Shift	0.7893 ± 0.0728	0.813 ± 0.0615	0.5850 ± 0.1115	0.6172 ± 0.1403	0.6439 ± 0.1036	0.6897 ± 0.1357
Gaussian Noise	0.8411 ± 0.0698	0.7663 ± 0.1032	0.7385 ± 0.1115	0.8016 ± 0.0525	0.5142 ± 0.1024	0.7323 ± 0.1446
Cut-out	0.7255 ± 0.0988	0.6955 ± 0.0648	0.6853 ± 0.0988	0.5444 ± 0.2082	0.4673 ± 0.1221	0.6236 ± 0.1590

In contrast, Cut-out augmentation resulted in the lowest average F1-score (0.6236), with particularly poor performance on FLD-N (0.5444) and TRA-N (0.4673). The sharp decline in these scores suggests that randomly removing portions of the data may have

disrupted key motion features, making it more difficult for the model to learn meaningful representations.

A two-way repeated measures ANOVA was conducted to examine the effects of augmentation methods and non-seen patterns on performance, using SPSS Statistics (version 29). A significant main effect of augmentation method was observed, $F_{(3,27)} = 16.295$, $p < 0.001$, with a large effect size ($\eta^2 = 0.644$), indicating that different augmentation methods resulted in significant differences in performance. Post-hoc comparisons using Bonferroni correction revealed that Gaussian Noise significantly outperformed the condition without augmentation ($p < 0.01$) and Cut-out ($p < 0.001$), while Time Shift also demonstrated a significantly higher F1-score than Cut-out ($p < 0.01$). These findings suggest that augmentation techniques, particularly Gaussian Noise, play a crucial role in improving model performance, whereas Cut-out resulted in the lowest accuracy.

These findings highlight the importance of selecting an appropriate augmentation method to improve model robustness. While Gaussian Noise and Time Shift contributed to notable performance gains, Cut-out led to a reduction in classification accuracy for certain patterns. The results suggest that augmentation strategies should be carefully tailored to the characteristics of the unseen dataset to maximize their effectiveness.

4. Discussion

The results demonstrate that the model was able to generalize well to unseen non-target movements while maintaining high classification accuracy for target patterns. This implies that the model may have effectively learned the characteristics of non-target movements, which often exhibit smooth curving trajectories. The use of Bi-LSTMs to capture long-term temporal dependencies in movement patterns and convolutional modules to extract local features may have contributed to a more balanced representation of both target and non-target patterns. Additionally, the target loss function introduced to distinguish between target and non-target patterns is likely to have been effective, leading to improved classification performance. However, the reduced recall for certain non-target patterns suggests that further refinement is necessary. Such refinement could enhance the model's capability to distinguish between target and non-target classes more effectively.

An analysis of the confusion matrices revealed that classification performance on non-target movements varied significantly depending on the excluded class. In most cases, non-target movements were frequently misclassified as target movements, indicating a challenge in distinguishing between these categories. This trend was particularly evident for the TRA-N pattern, which showed the lowest recall among non-target movements. A considerable number of TRA-N samples were classified as target movements, supporting the earlier observation that TRA-N is inherently difficult to separate from target gestures. This difficulty likely arises due to the unique circular trajectory of TRA, which differs from the directional shifts observed in other patterns. When TRA-N was excluded from the training data, the model struggled to establish a clear distinction, leading to a high misclassification rate. Consequently, the absence of this pattern in training data severely hindered the model's ability to generalize its classification criteria.

These findings highlight several key insights regarding the model's classification behavior. While the model demonstrates strong classification performance for most target patterns, it faces challenges in distinguishing non-target movements, particularly when they share trajectory similarities with target movements. The variation in confusion matrices across different experimental conditions underscores the impact of the excluded non-target pattern on overall classification performance. This suggests that the model relies on learned representations that are influenced by the specific set of training samples.

Table 5 compares the F1 scores of well-known state-of-the-art deep learning models and the proposed model for unseen non-target gesture classification. To adapt image-based models such as VGG16 and ConvNext for 1D data, the original 2D convolutional layers were replaced with 1D convolutions. Additionally, Gaussian augmentation was applied to all models to enhance their robustness.

Table 5. Performance comparison of state-of-the-art deep learning classification models for unseen non-target gesture classification. Gaussian augmentation was included for all the models.

Model	Ref.	Year	FLD-N	FLU-N	FRD-N	FRU-N	TRA-N	Mean $\pm$ Std
VGG16 (1D)	[37]	2015	0.6318 $\pm$ 0.1314	0.7461 $\pm$ 0.089	0.5704 $\pm$ 0.1278	0.6875 $\pm$ 0.0704	0.7222 $\pm$ 0.1148	0.6716 $\pm$ 0.1228
VGG19 (1D)	[37]	2015	0.698 $\pm$ 0.1495	0.6675 $\pm$ 0.1204	0.5703 $\pm$ 0.0835	0.6375 $\pm$ 0.1041	0.8231 $\pm$ 0.0508	0.6793 $\pm$ 0.1327
Inception (1D)	[38]	2016	0.6072 $\pm$ 0.095	0.659 $\pm$ 0.0983	0.7229 $\pm$ 0.0607	0.6845 $\pm$ 0.0949	0.7946 $\pm$ 0.0502	0.6936 $\pm$ 0.1013
Xception (1D)	[39]	2017	0.5792 $\pm$ 0.2001	0.6295 $\pm$ 0.1856	0.6824 $\pm$ 0.1281	0.7810 $\pm$ 0.0999	0.7756 $\pm$ 0.1219	0.6895 $\pm$ 0.1664
NasNet (1D)	[40]	2018	0.5625 $\pm$ 0.1719	0.6619 $\pm$ 0.1109	0.6961 $\pm$ 0.1034	0.622 $\pm$ 0.1516	0.6832 $\pm$ 0.0538	0.6452 $\pm$ 0.1296
EfficientNet (1D)	[41]	2019	0.6211 $\pm$ 0.1733	0.5048 $\pm$ 0.1062	0.6093 $\pm$ 0.1656	0.6656 $\pm$ 0.1234	0.7706 $\pm$ 0.1169	0.6343 $\pm$ 0.1597
ConvNext (1D)	[42]	2022	0.7019 $\pm$ 0.1119	0.5999 $\pm$ 0.137	0.557 $\pm$ 0.1186	0.6802 $\pm$ 0.0881	0.7601 $\pm$ 0.0699	0.6598 $\pm$ 0.1267
TimeMixer	[43]	2024	0.5678 $\pm$ 0.1154	0.5605 $\pm$ 0.1034	0.6011 $\pm$ 0.0729	0.6330 $\pm$ 0.1246	0.4042 $\pm$ 0.0813	0.5533 $\pm$ 0.1257
MordenTCN	[44]	2024	0.6583 $\pm$ 0.2536	0.7547 $\pm$ 0.0916	0.5970 $\pm$ 0.2104	0.7736 $\pm$ 0.1116	0.8272 $\pm$ 0.1174	0.7222 $\pm$ 0.1826
RobustTSF	[45]	2024	0.6457 $\pm$ 0.1156	0.7164 $\pm$ 0.0747	0.6895 $\pm$ 0.0825	0.7081 $\pm$ 0.1323	0.8515 $\pm$ 0.0592	0.7223 $\pm$ 0.116
Proposed		2025	0.8411 $\pm$ 0.0698	0.7663 $\pm$ 0.1032	0.7385 $\pm$ 0.1115	0.8016 $\pm$ 0.0525	0.5142 $\pm$ 0.1024	0.7323 $\pm$ 0.1446

Most traditional models demonstrated relatively lower performance in classifying FLD, FLU, FRD, and FRU compared to TRA. This trend indicates that conventional models tend to classify completely different patterns from the seen classes as out-of-distribution, while their classification accuracy decreases when encountering patterns that resemble the target gestures. This phenomenon suggests that models struggle to distinguish between non-target gestures that share similar characteristics with the target classes.

Furthermore, even the recently proposed models tailored for 1D signals, such as MordenTCN and RobustTSF, exhibited a similar tendency. RobustTSF, which is known for its robustness against noise and outliers, also showed reduced accuracy when classifying non-target gestures resembling the target classes.

The proposed model, however, achieved a 1.0%p higher average accuracy compared to RobustTSF. Unlike conventional models, the proposed approach demonstrated better performance in identifying unseen data that closely resemble target gestures. This improvement can be attributed to the model's architecture and the loss function designed specifically to handle such cases. On the other hand, the proposed model showed the lowest performance when classifying TRA among all models, indicating a limitation that warrants further research to improve model robustness in this aspect.

To further enhance classification performance, future improvements may involve integrating additional feature representations or refining training strategies. One potential approach could be increasing the diversity of non-target samples during training to improve the model's ability to generalize across unseen movements. Additionally, incorporating temporal dynamics more effectively through advanced sequence modeling techniques may help mitigate the observed misclassification trends.

Non-target data used in this study were intentionally designed to closely resemble target gestures while differing in subtle ways. This makes their classification inherently more difficult than distinguishing gestures that are entirely dissimilar. While gestures that are obviously distinct from the target set can often be detected using methods such as anomaly detection, our focus is on the more challenging case where structural similarities exist. Previous work, such as [29], has examined OOD classification with EMG-based

gestures that are clearly different from target actions. In contrast, the present study focuses on the harder scenario where non-target gestures exhibit similar spatial or temporal structures to the target classes. This setting reflects practical scenarios where non-target actions may not be entirely novel but rather slight variations of known gestures. The results demonstrate that the proposed approach effectively addresses this challenge. Nonetheless, we acknowledge the importance of evaluating model generalizability to completely novel gestures, which could be further explored in future work.

In addition, the limited number of participants posed another challenge to the model's generalizability. Deep learning models typically require large and balanced datasets to achieve robust performance. However, this study was constrained by a relatively small sample size—20 participants in the target group and 11 in the non-target group—which could potentially hinder generalization. To address this, the model architecture was intentionally kept compact to reduce complexity while preserving performance. Furthermore, the loss function played a key role in maintaining training effectiveness; one of its two components treated the output as a binary classification task, which helped stabilize learning even under class imbalance.

5. Conclusions

The classification of unseen non-target data refers to a model's ability to distinguish untrained movements that resemble common daily actions. This capability is essential in real-world gesture recognition systems, especially in asynchronous settings where unintended activations must be prevented.

To address this challenge, this study designed an experiment that focused on classifying unseen non-target movements using motion data. Unlike previous studies that primarily focused on target recognition or used conventional target gestures as non-target data, this study explicitly designed unseen non-targets: five non-target patterns corresponding to each of the five target movements. Through these experiments, we aimed to assess the model's ability to generalize classification criteria beyond the predefined training set. The proposed method was validated using arm movement data from 20 participants in the target group and 11 in the non-target group, achieving an average F1-score of 84.23%, with a non-target classification score of 73.23%.

The proposed approach demonstrated the ability to recognize untrained movements, enabling the model to adapt to novel actions and unexpected scenarios. The results also showed that non-target gestures resembling target gestures are prone to misclassification, which can be mitigated by data augmentation and a loss factor that emphasizes non-target filtering. Additionally, the choice of augmentation method is critical, as certain techniques may distort training signals and reduce performance on unseen data.

Future research should aim to improve the classification of unseen non-target patterns by addressing the limitations of training diversity. Although all unseen patterns in this study shared the characteristic of smooth movement, the proposed model struggled to distinguish some signals from target signals. A deeper understanding of this issue could be achieved by further investigating the impact of different movement patterns on classification performance. In addition, the non-target patterns used in this study were defined as smoothed versions of target movements, intentionally removing pauses and angular changes. While this design emphasizes subtle differences, it does not fully reflect the variability of natural movements that may be exhibited by real users. Future work should consider incorporating more diverse and unconstrained non-target gestures to better simulate real-world behavior.

Author Contributions: Conceptualization, W.-D.C.; Methodology, J.-H.C., H.-T.C., J.-S.J., S.-H.L. and W.-D.C.; Software, J.-H.C., K.-T.K. and W.-D.C.; Data curation, J.-S.J. and S.-H.L.; Writing—original draft, J.-H.C., H.-T.C., K.-T.K. and W.-D.C.; Writing—review & editing, W.-D.C.; Funding acquisition, W.-D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. RS-2024-00334159).

Institutional Review Board Statement: The study protocol was approved by the Institutional Review Board (IRB) of Pukyong National University (#1041386-202104-HR-12-01).

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Author Kyeong-Taek Kim was employed by the company Nexen Tire Research Center. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hsieh, C.-H.; Lo, Y.-S.; Chen, J.-Y.; Tang, S.-K. Air-Writing Recognition Based on Deep Convolutional Neural Networks. *IEEE Access* **2021**, *9*, 142827–142836. [\[CrossRef\]](#)
2. Hastie, T.; Simard, P.Y. Metrics and Models for Handwritten Character Recognition. *Stat. Sci.* **1998**, *13*, 54–65. [\[CrossRef\]](#)
3. Singh, D.; Khan, M.A.; Bansal, A.; Bansal, N. An Application of SVM in Character Recognition with Chain Code. In Proceedings of the International Conference Communication, Control and Intelligent Systems, CCIS 2015, Mathura, India, 7–8 November 2015; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2016; pp. 167–171.
4. Alqudah, A.; Alqudah, A.M.; Alquran, H.; Al-zoubi, H.R.; Al-qodah, M.; Al-khassaweneh, M.A. Recognition of Handwritten Arabic and Hindi Numerals Using Convolutional Neural Networks. *Appl. Sci.* **2021**, *11*, 1573. [\[CrossRef\]](#)
5. Bora, M.B.; Daimary, D.; Amitab, K.; Kandar, D. Handwritten Character Recognition from Images Using CNN-ECOC. *Procedia Comput. Sci.* **2020**, *167*, 2403–2409. [\[CrossRef\]](#)
6. Roy, P.; Ghosh, S.; Pal, U. A CNN Based Framework for Unistroke Numeral Recognition in Air-Writing. In Proceedings of the International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 404–409.
7. Sonoda, T.; Muraoka, Y. A Letter Input System Based on Handwriting Gestures. *Electron. Commun. Jpn. Part III Fundam. Electron. Sci. Engl. Transl. Denshi Tsushin Gakkai Ronbunshi* **2006**, *89*, 53–64.
8. Murata, T.; Shin, J. Hand Gesture and Character Recognition Based on Kinect Sensor. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*, 1–6. [\[CrossRef\]](#)
9. Kane, L.; Khanna, P. Vision-Based Mid-Air Unistroke Character Input Using Polar Signatures. *IEEE Trans. Hum. Mach. Syst.* **2017**, *47*, 1077–1088. [\[CrossRef\]](#)
10. Shin, J.; Kim, C.M. Non-Touch Character Input System Based on Hand Tapping Gestures Using Kinect Sensor. *IEEE Access* **2017**, *5*, 10496–10505. [\[CrossRef\]](#)
11. Saez-Mingorance, B.; Mendez-Gomez, J.; Mauro, G.; Castillo-Morales, E.; Pegalajar-Cuellar, M.; Morales-Santos, D.P. Air-Writing Character Recognition with Ultrasonic Transceivers. *Sensors* **2021**, *21*, 6700. [\[CrossRef\]](#)
12. Otsubo, Y.; Matsuki, K.; Nakai, M. A Study on Restoration of Acceleration Feature Vectors for Aerial Handwritten Character Recognition. In Proceedings of the Forum on Information Technology, Tottori, Japan, 4–6 September 2013; pp. 147–148.
13. Bamani, E.; Gu, P.; Zhao, X.; Wang, Y.; Gao, Q. Ultra-Range Gesture Recognition Using a Web-Camera in Human–Robot Interaction. *Eng. Appl. Artif. Intell.* **2024**, *132*, 108443. [\[CrossRef\]](#)
14. Chen, X.; Su, L.; Zhao, J.; Qiu, K.; Jiang, N.; Zhai, G. Sign Language Gesture Recognition and Classification Based on Event Camera with Spiking Neural Networks. *Electronics* **2023**, *12*, 786. [\[CrossRef\]](#)
15. Kuramochi, K.; Tsukamoto, K.; Yanai, H.F. Accuracy Improvement of Aerial Handwritten Katakana Character Recognition. In Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Kanazawa, Japan, 19–22 September 2017; pp. 116–119.
16. Matsuki, K.; Nakai, M. Aerial Handwritten Character Recognition Using an Acceleration Sensor and Gyro Sensor. In Proceedings of the Forum on Information Technology 2011, Hakodate, Japan, 7–9 September 2011; pp. 237–238.
17. Yanay, T.; Shmueli, E. Air-Writing Recognition Using Smart-Bands. *Pervasive Mob. Comput.* **2020**, *66*, 101183. [\[CrossRef\]](#)

18. Chamberland, F.; Buteau, É.; Tam, S.; Campbell, E.; Mortazavi, A.; Scheme, E.; Fortier, P.; Boukadoum, M.; Campeau-Lecours, A.; Gosselin, B. Novel Wearable HD-EMG Sensor with Shift-Robust Gesture Recognition Using Deep Learning. *IEEE Trans. Biomed. Circuits Syst.* **2023**, *17*, 968–984. [\[CrossRef\]](#)
19. Shin, J.; Miah, A.S.M.; Kabir, M.H.; Rahim, M.A.; Al Shiam, A. A Methodological and Structural Review of Hand Gesture Recognition Across Diverse Data Modalities. *IEEE Access* **2024**, *12*, 142606–142639. [\[CrossRef\]](#)
20. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *ACM Comput. Surv.* **2019**, *54*, 1–38.
21. Hsu, Y.; Shen, Y.; Jin, H.; Karaletsos, T. Generalized ODIN: Detecting Out-of-Distribution Image without Learning from Out-of-Distribution Data. *arXiv* **2020**, arXiv:2002.11297.
22. Fort, S.; Ren, J.; Lakshminarayanan, B. Exploring the Limits of Out-of-Distribution Detection. *NeurIPS* **2021**, *34*, 8830–8843.
23. Bontemps, L.; McDermott, J.; Koban, J.; Weber, J.; Muller, P.A. Deep Learning for Time Series Anomaly Detection: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 3434–3451.
24. Blázquez-García, A.; Conde, A.; Mori, U.; Lozano, J.A. Deep Learning for Anomaly Detection in Multivariate Time Series: Approaches, Applications, and Challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5002–5023.
25. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A.; Dau, H.A.; Lemaire, V.; Mueen, A.; Battenberg, E.; Keogh, E.; et al. Deep Learning for Time Series Anomaly Detection: A Survey. *Data Min. Knowl. Discov.* **2023**, *37*, 1521–1570.
26. Hong, Z.; Yue, Y.; Chen, Y.; Cong, L.; Lin, H.; Luo, Y.; Xie, S. Out-of-Distribution Detection in Medical Image Analysis: A Survey. *arXiv* **2024**, arXiv:2404.18279.
27. Feng, X.; Sha, H.; Zhang, Y.; Su, Y.; Liu, S.; Jiang, Y.; Ji, X. Reliable Deep Learning in Anomalous Diffusion Against Out-of-Distribution Dynamics. *Nat. Comput. Sci.* **2024**, *4*, 761–772. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Li, D.; Song, K.; Wang, J.; Zhang, Y. Time-Series Forecasting for OOD Generalization Using Invariant Learning. *arXiv* **2024**, arXiv:2406.09130.
29. Zhang, K.; Li, Q.; Chen, Y.; Liu, S.; Li, Z.; Liu, X.; Li, D. Diversify: A General Framework for Time Series OOD Detection and Generalization. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
30. Li, Y.; Liu, J.; Chen, Z.; Liu, X.; Wu, H.; Wang, L. Out-of-Distribution Detection via Deep Multi-Comprehension Ensemble. *arXiv* **2024**, arXiv:2403.16260.
31. Wang, Z.; Wang, S.; Xu, H.; Qiu, J.; Liu, C.; Zhou, Z. DACR: Distribution-Augmented Contrastive Reconstruction for Time-Series Anomaly Detection. *arXiv* **2024**, arXiv:2401.11271.
32. Xu, S.; Guo, M.; Zhao, X.; Liu, Y.; Hu, J.; Huang, Y. TS-OOD: Evaluating Time-Series Out-of-Distribution Detection and Prospective Directions for Progress. *arXiv* **2025**, arXiv:2502.15901.
33. Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; Xu, H. Time Series Data Augmentation for Deep Learning: A Survey. *arXiv* **2020**, arXiv:2002.12478.
34. Yang, H.; Desell, T. Robust Augmentation for Multivariate Time Series Classification. *arXiv* **2022**, arXiv:2201.11739.
35. Iwana, B.K.; Uchida, S. An Empirical Survey of Data Augmentation for Time Series Classification with Neural Networks. *PLoS ONE* **2021**, *16*, e0254841. [\[CrossRef\]](#)
36. Le Guennec, A.; Malinowski, S.; Tavenard, R. Data Augmentation for Time Series Classification Using Convolutional Neural Networks. In Proceedings of the ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Riva Del Garda, Italy, 19–23 September 2016.
37. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
38. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 2818–2826.
39. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 1251–1258.
40. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: New York, NY, USA, 2018; pp. 8697–8710.
41. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 6105–6114.
42. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 11976–11986.

43. Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J.Y.; Zhou, J. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 1–5 May 2024; pp. 1–12.
44. Luo, D.; Wang, X. ModernTCN: A Modern Pure Convolution Structure for General Time Series Analysis. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 1–5 May 2024; pp. 13–24.
45. Cheng, H.; Wen, Q.; Liu, Y.; Sun, L. RobustTSF: Towards Theory and Design of Robust Time Series Forecasting with Anomalies. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 1–5 May 2024; pp. 25–37.

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.