Computer Vision and Image Understanding 259 (2025) 104435

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Continuous hand gesture recognition: Benchmarks and methods

Marco Emporio [a], Amirpouya Ghasemaghaei [b], Joseph J. Laviola Jr. [b], Andrea Giachetti [a,*]

[a] University of Verona, Strada Le Grazie 15, 37134 Verona, Italy
[b] Department of Computer Science, University of Central Florida, Harris Engineering Center 321, Orlando, 32816-2362, FL, USA

ARTICLE INFO

Communicated by Juergen Gall

MSC:
41A05
41A10
65D05
65D17

Keywords:
Hand gestures
detection
classification
interaction

ABSTRACT

In this paper, we review the existing benchmarks for continuous gesture recognition, e.g., the online analysis of hand movements over time to detect and recognize meaningful gestures from a specific dictionary. Focusing on human–computer interaction scenarios, we classify these benchmarks based on input data types, gesture dictionaries, and evaluation metrics. Specific metrics for the continuous recognition task are crucial for understanding how effectively gestures are spotted in real time within input streams. We also discuss the most effective detection and classification methods proposed for these benchmarks. Our findings indicate that the number and quality of publicly available datasets remain limited, and evaluation methodologies for continuous recognition are not yet standardized. These issues highlight the need for new benchmarks that reflect real-world usage conditions and can support the development of best practices in gesture-based interface design.

1. Introduction

Gestural interfaces are becoming increasingly popular across various application fields and are expected to become even more wide spread in the near future. Given that gestures are the most common form of non-verbal communication, it is reasonable to rely on them as a foundation for building computer interfaces.

Enabling technologies are now inexpensive and easy to use: the most recent generation of Head Worn Displays provide hand tracking capabilities (Ungureanu et al., 2020; Schneider et al., 2021), and reliable software tools can perform real-time tracking of body limbs or fingers from RGB (D) streams (Zhang et al., 2020). Gesture-based control is already present in a variety of applications, such as virtual and mixed reality (Papadopoulos et al., 2021), industrial interfaces (Berg and Lu, 2020), automotive (Prabhakar and Biswas, 2021) and multimedia (Vatavu, 2012; Drossis et al., 2013) control, public kiosks (Huang et al., 2020), making it possible to interact with computers naturally without the need for input devices. For this reason, in recent years, relevant efforts have been dedicated to gesture recognition research, and several authors have recently published bibliographic surveys aiming to analyze and classify the published works (see Section 2.1). However, when looking at the literature, it is possible to see that most of the algorithms presented and classified in the existing surveys are not designed to detect and recognize gestures in a *continuous* stream of data; they instead aim only to address the problem of classifying segmented gestures.

Many popular benchmarks widely used to evaluate gesture recognition methods only provide recordings of segmented gestures for training/testing of algorithms and evaluating classification accuracy; some examples of these benchmarks include VIVA (Ohn-Bar and Trivedi, 2014), DHG 14–28 (De Smedt et al., 2016), SHREC'17 (De Smedt et al., 2017), and Jester (Materzynska et al., 2019). It is worth noting that for some popular benchmarks, such as EgoGesture (Zhang et al., 2018), featuring both a continuous detection and a segmented classification, the latter is significantly more popular.

This fact is surprising since it is more often necessary, in practice, to design methods to solve the so-called "continuous" or "online" gesture recognition problem, requiring the detection of the gesture in the input stream. This problem is a particular case of event detection, in which significant gestures are interleaved with non-significant hand movements ("non-gestures"), and the recognition algorithms must segment the gestures and classify them correctly, providing feedback with low latency. The algorithms should minimize false positives and false negatives to avoid activating unwanted actions.

Recognition delay can be critical in applications such as extended reality (XR), gaming, or robot control in hazardous environments. However, offline benchmarks are not suitable for evaluating this delay, as they assume gestures are pre-segmented and do not consider the exact moment when the method produces a prediction. Consequently, they fail to indicate when the system recognizes a gesture relative

* Corresponding author.
E-mail address: andrea.giachetti@univr.it (A. Giachetti).

https://doi.org/10.1016/j.cviu.2025.104435
Received 10 April 2024; Received in revised form 23 May 2025; Accepted 24 June 2025
Available online 2 July 2025

Table 1
Comparison of the key characteristics of other survey papers on hand gesture recognition.

Survey	Scope	Discussed		
		Benchmarking	Continuous recognition	Continuous evaluation
Mitra and Acharya (2007)	Hand, body, and face gestures	✗	✗	✗
LaViola (2013)	3D gestures	✗	✗	✗
Rautaray and Agrawal (2015)	Methods in vision-based recognition	✗	✗	✗
Pisharady and Saerbeck (2015)	Recognition methods and databases	✓	✗	✗
Cheng et al. (2016)	3D gesture datasets	✓	✓	✗
Kakkoth and Gharge (2017)	Real-time gesture recognition	✓	✗	✗
Escalera et al. (2017)	Multimodal gesture/action recognition	✓	✗	✗
Gu et al. (2021)	Action recognition datasets and evaluation	✓	✗	✗
Oudah et al. (2020)	Classification through task characteristics	✓	✗	✗
Shi et al. (2021)	Deep learning on gesture videos	✓	✗	✗
Sarma and Bhuyan (2021)	Methods in vision-based recognition	✓	✗	✗
Jain et al. (2022)	Methods in vision-based recognition	✓	✓	✗
Our Survey	**Continuous hand gesture recognition**	✓	✓	✓

to its actual occurrence, making it impossible to measure whether recognition is early or late.

Evaluating an algorithm for continuous gesture recognition is not trivial, and it should consider all the aforementioned aspects, not only the classification accuracy. Existing works that survey gesture recognition algorithms, as discussed in Section 2.1, are not focused on the continuous problem, so they are not helpful for an interaction designer searching for guidelines to create user interfaces based on gestures.

In this work, we try to fill this literature gap by proposing an updated state-of-the-art report focused on the *continuous* recognition of *intentional* gestures for *Human–Computer Interaction*, namely for communication and system commands, using off-the-shelf technology. The base of our work is a systematic search for specific benchmarks to evaluate this task. We classified the benchmarks found according to a taxonomy describing their characteristics, and, for each of them, we looked at the methods providing the best results according to the specific evaluation metrics used, analyzing the main features of the gesture detection methods proposed. The paper is structured as follows: Section 2 motivates our work by analyzing existing survey papers on hand gesture recognition, highlighting that none of them adopt a focus similar to ours. The section also describes the methodology adopted to create our new survey. Section 3 introduces the criteria used to classify the research papers we reviewed: gesture capture technology, input data encoding, the types of gestures included in the dictionary, and the metrics used to compare the tested algorithms. Additionally, we provide a taxonomy of the different gesture recognition approaches applied in these studies. Section 4 presents the identified benchmarks, classifying them according to the previously introduced taxonomies and highlighting the techniques that achieved the best results. Finally, Section 5 presents a discussion of the survey's findings, identifying open issues, and suggesting research directions for future work.

2. Survey motivation and methodology

Several surveys on hand gesture recognition analyze the existing literature, but do not focus on continuous gesture recognition and related benchmarking issues. This gap motivated us to conduct a novel analysis of the literature, concentrating on these particular aspects through the methodology detailed in Section 2.2.

2.1. Existing surveys on hand gesture recognition

Searching for existing surveys on hand gesture recognition, we found that some of them are outdated and generic, like the work of Mitra and Acharya (2007). This survey is not limited to hand gestures but also deals with body and face movements.

The survey of LaViola (2013), addresses many research challenges but does not discuss the benchmarking of continuous gesture recognition. Rautaray and Agrawal (2015), Focus on vision-based gesture recognition. They also analyze commercial products for gesture tracking, but do not address the issues related to validation and benchmarking.

Pisharady and Saerbeck (2015), presents a review that includes an analysis of both methods and databases that appeared in the gesture recognition literature. Here, the evaluation of continuous detection is also not considered. Cheng et al. (2016), deals with 3D gesture datasets and introduces the problem of continuous gesture recognition. However, they do not discuss the related evaluation or analyze the existing continuous gesture detection benchmarks.

Kakkoth and Gharge (2017), also focuses on real-time hand gesture recognition and provides a taxonomy of hand gestures (dynamic and static) and detection methods (sensor-based, vision-based, and depth-based). However, the description is short, and benchmarking and continuous recognition are not considered. Escalera et al. (2017), presents a comprehensive overview of the challenges in multi-modal gesture (and action) recognition, discussing the outcomes of several contests in the domain. The paper is one of the few that discusses the evaluation of recognition algorithms. However, most of the tasks proposed in the contests are not continuous, and the discussed methods are nowadays outdated.

Gu et al. (2021), also deal with datasets and evaluation, but their survey is mostly dedicated to generic action recognition and not to gesture recognition. The evaluation methods presented are not appropriate for continuous gesture recognition. Oudah et al. (2020), classifies a considerable number of research papers according to different recognition tasks' characteristics, but does not address the problems of benchmarking and continuous detection. Shi et al. (2021), focuses on gesture recognition in videos based on deep learning. The paper presents a list of gesture recognition benchmarks but does not clearly define a taxonomy or distinguish between segmented and continuous recognition. The authors also do not discuss the evaluation approaches.

Sarma and Bhuyan (2021), presents a detailed description of the methodologies applied in vision-based hand gesture recognition. However, they dedicate little space to benchmarks and evaluation. Jain et al. (2022), presents several benchmarks and methods, mixing continuous and non-continuous ones without discussing the differences in the evaluation. The paper focuses on techniques used for video-based gesture classification.

While the high number of surveys demonstrates the growing interest in the topic, it is difficult to find useful and up-to-date information to design interactive systems based on gestures captured with modern sensors. As shown in Table 1, we did not find surveys dealing with the evaluation of continuous gesture recognition for interaction tasks, discussing all the related issues in detail. For this reason, we decided to survey recent works focusing on this particular domain. The idea is to start from benchmarks and related evaluation metrics, trying to understand the peculiarities of continuous recognition, the different approaches available for the specific evaluation, and the effectiveness of the different detectors proposed for the related tasks.

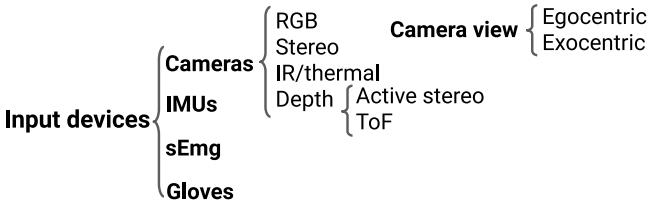

Fig. 1. Classification based on input types.

2.2. Literature search methodology

To search for the benchmarks with the required features (with a focus on continuous recognition of mid-air hand and upper body gestures), we first used the websites that are popular for benchmarking initiatives, such as Eurographics Shape Retrieval Contests (SHREC),[1] ChaLearn,[2] and Kaggle,[3] this resulted in 3 relevant benchmarks. We then conducted methodical research in generic literature archives, following the guidelines of the PRISMA 2020 methodology (Page et al., 2021).

This procedure started with the "identification" phase aimed to form a superset of the papers that are related to "gesture recognition benchmarks and dataset" and "evaluation metrics".

We defined a base *BASE* query as follows:

```
BASE= {(Hand OR (Upper and Body)) AND (Gesture OR Gestural OR Interaction OR Interactions OR Recognition) AND (Benchmark OR Benchmarks OR Dataset OR Datasets OR (Data and Set) OR (Data and Sets)) AND (Online OR Continuous OR Real-time OR (Real and Time))}.
```

We performed this search on Scopus[4] since it is reliable, presents more relevant results, and has advanced searching capabilities that include the usage of custom queries. The query resulted in 2758 articles, and to find the most relevant ones, we initiated a screening phase in which we carefully read all the abstracts, keeping only the ones that matched our inclusion criteria, defined as follows:

- The paper must describe a benchmark for gesture recognition in continuous time.
- The gestures should be designed for interaction tasks, which are intentional, voluntary gestures used to control a system, issue commands, or communicate with a digital interface. This includes command and communication gestures commonly used in human–computer interaction scenarios. In contrast, gestures that are not explicitly intended for interaction, such as those used in general action recognition (e.g., walking, running) or in sign language recognition, are excluded.

We then thoroughly read the filtered papers to ensure that only the relevant ones are kept. Only 13 benchmark papers matched our strict inclusion criteria, and we excluded a benchmark (SHREC '21, Caputo et al., 2021) as the benchmark's authors declared annotation issues on their website. The benchmarks found were classified and analyzed according to several criteria, introduced in Section 3, and related to the acquisition method, the data encoding, the gesture dictionary, and the evaluation metrics.

Furthermore, we investigated the recognition approaches used on the benchmarks, introducing a specific taxonomy of the methods and analyzing the ones providing the best results on each benchmark. For this purpose, we conducted a backtracking search of the papers that cited the related papers/sites. This search was performed on Scopus and Google Scholar[5] and resulted in 1645 papers. From them, we selected only the works that tested novel methods on the benchmarks' data, and, for those providing the best results (up to three per benchmark), we give a short description, classifying their characteristics according to our taxonomy and present the performance metrics in Section 4.

3. Benchmark taxonomy

We classified the selected benchmarks according to a set of characteristics that need to be considered to understand the practical applications where they can be exploited.

- **Input data**: the characteristic of the sensing devices and the acquisition setup.
- **Data encoding**: The provided data stream to train and test the detection frameworks.
- **Gesture set**: the dictionary of gestures included in the benchmarks.
- **Evaluation metrics**: The measurements used to assess the effectiveness of the detection methods for the specific task.

Furthermore, we also classify the recognition approaches used on them.

3.1. Input modalities and data stream encoding

Devices (summarized in Fig. 1) used to capture gestures belong to two main categories: cameras and wearables. In the first category, we have **RGB cameras**, e.g., webcams in desktop interfaces, or devices mounted on HMDs for XR applications. **Depth sensors/RGBD cameras** are also used to capture gestures. Depth information facilitates the spatial segmentation of arms/hands. Among the various cheap depth sensing devices used in gesture-based interfaces, we can mention **stereo cameras** (e.g., Zed, UltraLeap), **active IR sensors** (e.g., Microsoft Kinect, Intel Realsense De Smedt et al., 2016), **Time-of-Flight (ToF)** cameras (e.g., the one mounted on HoloLens). **IR/thermal cameras** have also been proposed for gesture capture (Vandersteegen et al., 2020). Camera-based acquisition is simple and non-invasive, but it can suffer from occlusions, which may cause degradation of the data quality (Lee et al., 2022).

This issue is strictly related to another critical design choice for the acquisition, that is, where to place the sensors with respect to the subject's body/hands. The choice of viewpoint depends on the target application and can influence the performance of the recognizers. Our taxonomy defines two categories for the camera viewpoint: **Egocentric**, e.g., captured with head-mounted cameras, and **Exocentric**, e.g., captured using external cameras. Egocentric benchmarks are useful to test recognizers for XR applications. Depending on the sensor distances, exocentric benchmarks are suitable for designing methods used in Human–Robot interaction, public kiosks, multimedia, or automotive controls.

Wearable sensors are typically more invasive than external cameras but are unaffected by occlusions. **Inertial measurement units (IMUs)** can provide data such as acceleration, angular rate (rotation speed), and orientation in space, and have been proposed in benchmarks for gesture recognition (Ruffieux et al., 2013). **VR gloves** can also be equipped with sensors to capture joints' motion (Caeiro-Rodríguez et al., 2021), but are expensive and rarely used in real-world interaction scenarios. We did not find benchmarks with glove-based data. Surface Electromyography (**sEMG**) based wearable devices like the Myo armband (Benalcázar et al., 2017) have been recently used to create systems that can recognize dynamic hand gestures. We did not find public benchmarks based on them for continuous gesture recognition, while examples with segmented ones are available (Atzori et al., 2014; Amma et al., 2015).

[1] https://www.shrec.net
[2] https://chalearnlap.cvc.uab.cat/
[3] https://www.kaggle.com
[4] https://www.scopus.com/home.uri
[5] https://scholar.google.com/

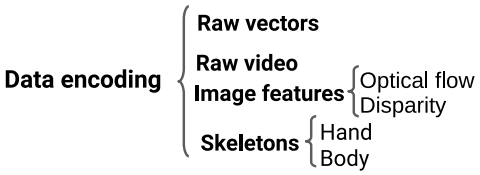

Fig. 2. Classification based on stream encoding.

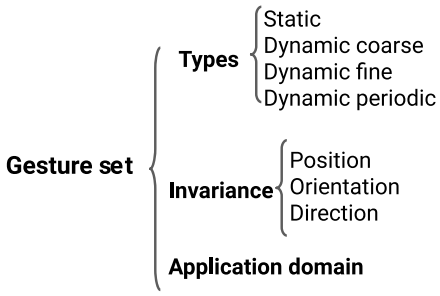

Fig. 3. Classification based on gesture dictionaries.

3.2. Input stream encoding

The main types of data streams (shown in Fig. 2) processed by gesture detection modules are **raw video data, visual features (optical flow, disparity), arm/hand skeleton sequences, and raw vectors of wearable sensors' data**. Benchmarks provide the datasets in one or more of these encodings.

The most popular solution is to use raw sequences as recognizer input, meaning that the tested technique needs to perform both temporal and spatial segmentation of the hands. NVGesture (Molchanov et al., 2016) provides dense optical flow and disparity maps as additional input.

Skeleton data can be a reasonable input for recognition, as they are directly provided by the API associated with gloves or body/finger-tracking devices. Inexpensive specialized devices like the Ultraleap products provide real-time hand skeleton data at high frame rates, and finger tracking modules are integrated in popular XR headsets like Hololens 2 or Meta Quest and used for gesture-based interaction. The skeleton representation typically includes the 3D spatial coordinates of joints and possibly quaternions representing segments' orientation. Hand skeletons could be estimated from videos using specialized software tools like Google MediaPipe (Zhang et al., 2020). The skeleton representation is compact and filters out non-relevant data captured by the original sensor. The drawback is related to the possible failure of the hand pose estimation algorithm. Several gesture-recognition benchmarks (De Smedt et al., 2017; Caputo et al., 2019; Emporio et al., 2022) include hand skeleton data.

In the case of non-spatial acquisition techniques (IMUs, sEMG), the detector works directly on the stream of the captured **raw data**. In this case, additional helpful information for gesture recognition may be derived from the position of different worn sensors (Ruffieux et al., 2013).

3.3. Hand gestures' taxonomy and gesture dictionaries

In the tasks of our interest, according to the taxonomy of Karam and Schraefel (2005), we consider only "semaphoric" or "language" gestures that we must distinguish from "gesticulation" or non-gesture movements. We selected several aspects to characterize the different sub-types of these gestures found in the research papers of our interest (as illustrated in Fig. 3). The first one is related to their temporal features. For this aspect, we use a taxonomy similar to the one proposed in Li et al. (2019a), distinguishing **Static** and **Dynamic** gestures. Static gestures are those where the semantic is determined by the hand pose (spatial posture of the joints), without significant movements. A static gesture is performed by holding a specific hand configuration or pose for a minimum duration. Dynamic gestures are those where the semantic depends on the movements of the hands and can be further divided into up to three sub-types, as done in some existing benchmarks such as in Emporio et al. (2022):

- **Dynamic Coarse**, characterized by the hand's global trajectory, for example, drawing a cross in mid-air.
- **Dynamic Fine**, where the semantic depends on a variation in the fingers' articulation over time, such as pinch or grab.
- **Dynamic periodic**, where the semantics depend on the iteration of a basic motion pattern such as waving a hand or a finger.

The semantics of a gesture can be invariant with respect to specific changes in its execution. Invariance properties of the gestures are important as they influence the design of proper recognition tools. We can distinguish the following cases:

- **Position-invariant** gestures are those where a translation does not change the semantics. Interfaces where some gestures are considered meaningful only if executed in specific spatial regions employ non position-invariant dictionaries.
- **Orientation-invariant** gestures are those where a global hand rotation does not change the semantics. For example, a gesture dictionary that includes a thumb up or down gesture to indicate like or dislike is clearly non-invariant to orientation.
- **Direction-invariant** gestures are the dynamic ones where the correct labeling does not depend on the orientation of the global movements of the hand with respect to a fixed reference frame. The pinch gesture used in many practical VR interfaces to trigger actions has the same meaning, independent of the global hand movement.

To allow the users to interact with gesture-based interfaces, the designers (and the creators of benchmarks) create **dictionaries** of gestures, possibly including static and dynamic ones. In our classification of gesture dictionaries, we will consider Position (Orientation, Direction) invariant, a dictionary of gestures where all the items feature the same property so that the recognizer can be trained/tested with an invariant encoding.

Dictionaries are characterized by the sets of class labels and by the origin of their semantics. Dictionaries may assign the semantics based on simple rules, such as error margins with respect to a standard pose/trajectory template. In this case, the classification errors may depend only on finger-tracking accuracy. In the reviewed benchmarks, the semantics are always data-driven and depend on a number of training templates or sequences. This can make the task challenging as different users can execute the gestures in quite different ways because of constraints in the individual hand articulation (Lee and Jung, 2015), different speeds, and different mental models of the gestures, possibly due to different cultural backgrounds. Hand morphology and size can vary, and the skin color can differ, and this can create challenges for image-based detection.

The **number** and the **diversity** of the subjects performing the gestures of the datasets are, therefore, important characteristics to understand how much the results obtained on a benchmark can be generalized in a practical setting. The limited ability of the training examples to represent the target users can be a relevant issue in practical applications, requiring the collection of a huge amount of additional training data. The **choice of the gesture classes** in the dictionary is also a critical feature of the benchmark to be pointed out. Dynamic ones can have an extremely variable time duration, quite challenging for online recognizers, and different gestures can share similar subparts. The recognition rates for individual gestures may depend critically on

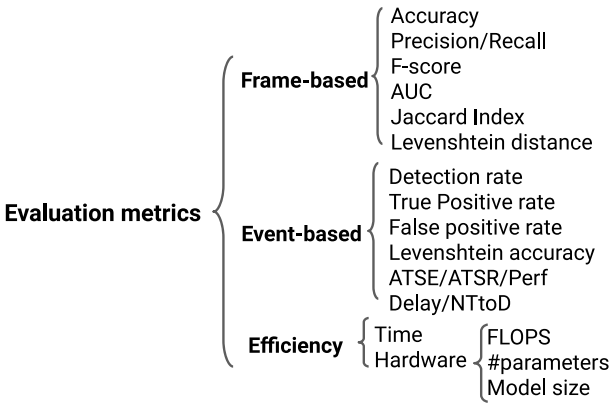

Fig. 4. Classification based on evaluation metrics.

the other ones included in the dictionaries, which might be optimized for the discriminability of gestures. The dictionaries of the benchmarks included in our survey do not seem to have been optimized in this regard.

The **number of sequences** provided for training (and testing) is another relevant characteristic of the benchmark, as well as the type of gesticulation done in the non-gesture parts. In fact, the possible use and the performance of machine learning tools may be strongly affected by the low number of training examples. This problem can be mitigated by data augmentation (Maghoumi et al., 2021; Cabrera and Wachs, 2018), but with some limitations in the case of continuous recognition. Meaningful gestures of a selected dictionary cover a small part of the potential "gesture space", including all the possible hand position/orientation/pose variations. This means that in an online classifier where we need to discriminate the gestures from the "non-gesture" class, we have a large imbalance in the "gesture space" partitioning, with most of the space labeled as "non-gesture".

Training data for the continuous gesture recognition method are not (or not only) segmented gestures, but long sequences where the time location of the meaningful gestures is given as data annotation. The characteristics of the non-gesture patterns used to interleave the gestures in these sequences should be accurately controlled to avoid unintentional executions of gestures not annotated. Ideally, the non-gesture parts should represent all the gesticulations done by all the users of the interactive system simulated, optimally sampling the "non-gesture" space.

In the case of video data, the diversity of the visual scenarios for the data capture is also important to assess the robustness of the detector against changes in background, light conditions, and others. Therefore, we report the number of different scenes used in the datasets. In the practical design of the acquisition devices and the dictionary, many choices strongly depend on the application focus. We introduce a specific category in our taxonomy to record if a reviewed benchmark has a specific **application domain**. Specific applications for which continuous mid-air gesture recognition systems can be applied are mixed reality, automotive, public kiosks, sign language-based communication, and mobile/touchscreen interfaces.

3.4. Evaluation metrics

As discussed in Section 1, while for the segmented gesture classification task, the evaluation is simple and based only on the test gesture labeling accuracy, for the continuous classification task, the performance assessment is more complex, as it should also consider the temporal segmentation quality, the latency, and the probability of false detection.

Following Ward et al. (2011), (as show in Fig. 4) we divide the metrics into two groups:

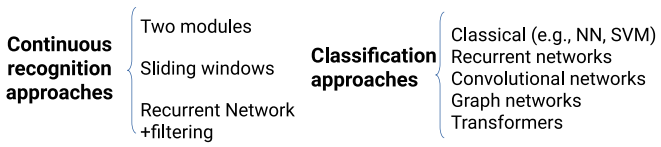

Fig. 5. Classification of continuous gesture spotting/recognition methods.

- **Frame-Based.** These metrics are derived from the per-frame comparison of the predicted labels with the ground truth ones. Examples are the average classification scores of the single-frame labeling (Accuracy, Precision/Recall, F-Score, Area under curve (AUC)) or string distances (Jaccard Index, Goswami et al., 2018, Levenshtein Distance, Levenshtein et al., 1966).
- **Event-Based.** A gesture in a continuous sequence of frames can be considered an *event* with a specific start time, end time, and an associated label. An evaluation based on these parameters should cope with the temporal location of the gestures, possibly estimating their accuracy, and must define thresholds to consider whether an event is correctly detected. While frame-based metrics compare two strings of equal length with ground truth and the estimated sequences of frame labels, event-based metrics evaluate the differences between ground truth and estimated lists of events with associated attributes. These lists may have different lengths, as we can have missed and falsely detected events, so that the comparison is not straightforward. For these reasons, various metrics have been proposed and used in different benchmarks, making it difficult to compare the results obtained.
  The **Detection Rate** (percentage of events correctly captured with respect to the ground truth total number, Ruffieux et al., 2013; Caputo et al., 2019; Emporio et al., 2022) and the **False Positives Score** (percentage of false detections with respect to the ground truth total number, Caputo et al., 2019; Emporio et al., 2022; Molchanov et al., 2016; Wannous and Vandeborre, 2022), are often measured, even if with variable criteria to consider a gesture correctly detected. Another viable option is the **Levenshtein Accuracy** (Benitez-Garcia et al., 2021b; Xu et al., 2023), which is based on comparing strings of consecutive events.
  To characterize the accuracy of the temporal segmentation of the methods, some authors use the delay of the recognition feedback. In Ruffieux et al. (2013), new specific scores are proposed for this purpose: the **Accurate Temporal Segmentation Rate (ATSR)**, the **Absolute Temporal Segmentation Error (ATSE)**, and the **Performance Index**. The computation of ATSR is derived by summing the absolute temporal discrepancies between the algorithm's prediction and the ground truth for both the start and stop events. The result is then divided by the total duration of the gesture occurrence. The Performance Index (Perf) is a single metric that combines F-score and ATSR.

In our taxonomy, we also consider the **efficiency metrics**, to evaluate how much an algorithm can be adapted to different platforms, as many benchmarks also assess the amount of resources needed by the methods in terms of memory requirements (e.g., the number of parameters of the neural models used, overall model size), computational complexity and real-time detection performances (e.g., FLOPS).

3.5. Gesture detection methods

Fig. 5 shows a taxonomy of the different continuous recognition approaches that have been applied to the data and tasks of the surveyed benchmarks. Even if it is not an intrinsic property of the benchmarks, this classification can reveal how well the different recognition approaches perform on them, given the best results presented in Section 4. The methods can be classified according to how they deal with the

Table 2
Characteristics of the gesture dictionaries used in the reviewed benchmarks: types of gestures (S = static, DC = dynamic coarse, DF = dynamic fine, DP = dynamic periodic) and invariance (P = position, O = hand orientation).

Name	Year	Type of gestures	Invariance	Application domain
ChAirGest	2013	DC, DP	P	General interaction
Montalbano T3	2014	DF, DP	P	Verbal communication
ConGD	2016	S, DF, DC, DP	P	Many domains
NVGesture	2016	S, DF, DC, DP	P	Automotive
EgoGesture	2018	S, DF, DC, DP	P	Mixed reality
SHREC'19	2019	DC	P	Mixed reality
IPN Hand	2020	S, DF, DC	P, O	Touchless screens
MlGesture	2020	DC,DP	P, O	Automotive
LD-ConGR	2022	S, DC	P, O	Long distance recognition
SHREC'22	2022	S, DF, DC, DP	P	Mixed reality
ODHG 14/28	2022	DF, DC	P, O	General interaction
ZJUGesture	2023	DF, DC	P, O	Mobile

Table 3
Characteristics of the acquisition setups for the datasets included in the reviewed benchmarks.

Name	Data types	View	Input devices	FPS
ChAirGest	RGB, RGB-D, Hand pose, IMU	EGO, EXO	Kinect/Xsens MTw IMU	30/50
Montalbano T3	RGBD	EXO	Kinect	20
ConGD	RGB-D	EXO	RGB-D Camera Setup	NA
NVGesture	RGB-D, Optical flow, IR	EXO	SoftKinetic DS325/DUO 3D	30
EgoGesture	RGB-D	EGO	Intel RealSense SR300	30
SHREC'19	Hand pose	EGO	Leap Motion	NA
IPN Hand	RGB	EXO	Webcam	30
MlGesture	RGB-D, RGB, Thermal	EXO	Industrial cameras	8/12/16
LD-ConGR	RGB-D, RGB	EXO	Kinect v4	30
SHREC'22	Hand pose	EGO	Hololens 2	20
ODHG 14/28	RGB-D, Hand pose	EXO	Intel Real-Sense	30
ZJUGesture	RGB	EXO	Mobile phone	30

Table 4
Characteristics of the sets of sequences/gestures included in each benchmark. Figures in brackets specify the training/test splitting of sequences and gesture samples. Gps means Gestures per sequence (V = Variable).

Name	Scenes	Subjects	Classes	Gesture samples	Sequences	Gps
ChAirGest	1	10	10	1200 (900 – 400)	NA	V
Montalbano T3	1	10	10	13,858 (3362 – 2742)	563	V
ConGD	15	21	249	47,933 (30,442 – 17,491)	22,535	V
NVGesture	1	20	25	1532 (1050 – 482)	1532	V
EgoGesture	6	50	83	24,161 (19,184 – 4977)	2081	9 to 12
SHREC'19	1	13	5	195 (60 – 135)	195	1
IPN Hand	28	50	13	4218 (148 – 52)	200	1 to 5
MlGesture	1	24	9	over 1300	Over 1300	1 to 2
LD-ConGR	5	30	10	44,887 (34,315 – 10,572)	542	V
SHREC'22	1	6	16	1152 (576 – 576)	288	3 to 5
ODHG 14/28	NA	20	28	2800	NA	V
ZJUGesture	12	60	9	9892 (8290 – 1602)	NA	V

continuous task, performing both temporal segmentation (also spatial in the case of video input) and classification of gestures.

A possible solution is to use **two modules**: one designed to perform the time segmentation, and one for the classification. A typical solution for the segmentation is to estimate the boundaries based on the motion energy (Kahol et al., 2003; Li et al., 2019c). The subsequent classification can be done with classifiers trained with labeled segmented gestures (of different sizes). Another option is to use a **sliding windows** approach in the time domain, classifying one or more signal windows (Dietterich, 2002) ending at the current time frame with a specifically trained classifier to determine the frame label.

When using hand pose data, the classification module can be based on simple template matching (Caputo et al., 2018), but the recent literature is dominated by neural networks. Different network architectures have been proposed, such as 1D convolutional networks (Yang et al., 2019), graph-convolutional networks (Li et al., 2019b), or transformer networks using the attention mechanism to focus on the co-occurrence of relevant features of the sequences in gesture classes (Shi et al., 2020). 2D/3D convolutional networks can be applied for RGB-D images (Tran et al., 2015). Another possible solution for the continuous recognition is the use of a **recurrent network**, trained on continuous data and providing a predicted gesture label at each time step, using non-gesture as an additional class (Tsironi et al., 2017; Chai et al., 2016; Maghoumi and LaViola, 2019). In this case, the preliminary segmentation of the gestures in the test phase is unnecessary, but a further post-processing step is required to derive a reliable detection of gesture events from the frame labels.

4. Surveyed benchmarks

This section presents the main characteristics of the 12 benchmarks selected, classified according to our taxonomy. These characteristics are summarized in Tables 2, 3, and 4. In detail, Table 2 presents the characteristics of the gesture dictionaries included, Table 3 shows the characteristics of the acquisition setups, and Table 4 presents more details of the datasets collected.

For each benchmark, we also describe the methods that provide the best results on it, following the criteria described in Section 3.5.

4.1. ChAirGest

The ChAirGest challenge (Ruffieux et al., 2013) was an open research initiative aimed at motivating scholars to leverage data collected

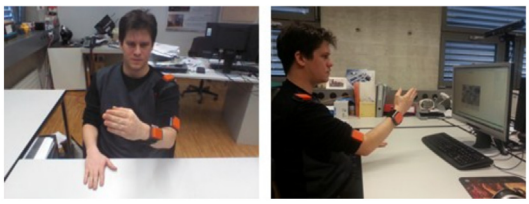

Fig. 6. Recording setup for the ChAirGest Benchmark. Left: image captured by the Kinect. Right: external view of the same subject. Wearable IMUs are visible on his right arm.
Source: From Ruffieux et al. (2013).

Table 5
Performances of the best method on the ChAirGest benchmark.

Method	Date	ATSE	ATSR	F1 score	Perf
CHMM	2013	NA	0.923	0.907	0.912

Fig. 7. Example gestures from the Montalbano dataset. They are performed before a Microsoft Kinect while speaking Italian.
Source: From Escalera et al. (2015).

Table 6
Performances of the best methods on the Montalbano Track 3 benchmark.

Method	Date	JI (%)
ConvNetsFusion	2021	92.3
Temp Conv + LSTM	2016	90.6
RNN + LSTM Cells	2016	88.8

from diverse sensors to enhance and assess methods for gesture spotting and recognition. It was based on a dataset created using a Kinect RGBD camera (30 fps) and four IMUs (50 fps), strategically attached to the right arm and neck of ten subjects (Fig. 6). The dataset consists of a vocabulary of 10 one-hand dynamic gestures commonly used in close human–computer interaction (e.g., swipe, circle, push, etc.), initiated from three different resting postures and captured across two distinct lighting scenarios of the same scene. The authors collected a total of 1200 annotated gestures, organized into continuous video sequences, each containing a variable number of gestures. The evaluation is based on a hybrid approach incorporating well-established metrics (event-based precision and recall and the derived F-score) and an original time-based metric, the Accurate Temporal Segmentation Rate.

4.1.1. Best results on ChAirGest

Table 5 shows the metrics obtained with the best method tested on the ChAirGest benchmark, **Concatenated Hidden Markov Models (CHMM,** Yin and Davis, 2013). CHMM is based on three core components: a feature extraction module, a temporal segmentation module, and a gesture spotting and recognition module. In the first step, for classification, a 12-dimensional feature vector is extracted from the Xsense data (linear acceleration, angular velocity, and Euler orientation) and Kinect data (relative position between shoulder and hand). The hand position given by the Kinect APIs was not considered to be reliable, and a custom salience-based hand segmentation method based on the RGB-D data was applied to replace it. The gesture spotting method exploits the training data from all the users to create Gaussian models for the 'rest positions' and a Gaussian model for 'non-rest positions', and then classifies the frames accordingly. Gestures are recognized when consecutive non-rest positions are detected for over 0.25 s. The gesture recognition exploits multiple Hidden Markov Models (HMM) trained for the three main gesture phases: pre-stroke, nucleus, and post-stroke. The three HMMs are concatenated to form one HMM for each gesture. To detect and correctly segment the nuclei of the gestures, the Viterbi algorithm is applied to find the most probable hidden state sequence.

4.2. Montalbano track 3

The Looking at People Challenge 2014 competition (Escalera et al., 2015) included three different tracks and related benchmarks. The first two focus on action recognition and cannot be included in our survey; only the third complies with our inclusion criteria. In this track, the participants were tasked with recognizing 20 distinct Italian gesture categories using continuous RGB-D video sequences captured at 20 fps. The dataset comprises 13,858 gestures performed by 27 subjects in 563 sequences. From those sequences, 287 were given to the participants as training sequences and labels; the other 276 were used as test sequences. Throughout all the sequences, a single user is captured standing before a Kinect device without occlusion. The gestures in this dataset are not "commands" but are "natural" communicative gestures performed while the subjects were speaking fluent Italian (Fig. 7). Each sequence is 1–2 min long, and during it, the subjects move their arms without rest, resulting in an extremely varied "non-gesture" class. This dataset exhibits a notable intra-class variability among gesture samples, meaning that gestures within the same category can vary significantly. Conversely, the inter-class variability is relatively low, indicating that some gesture categories share similarities, making the classification hard. It is worth noting that no information is available regarding the number of gestures included within each sequence. The benchmark uses the Jaccard Index to evaluate the recognition's accuracy.

4.2.1. Best results on Montalbano track 3

Table 6 shows the metrics obtained with the best results applied to this benchmark. **ConvNetsFusion** (Wang, 2021), is a two-module method. The temporal segmentation module combines two methods: the first detects the intervals where the height of the hands, segmented with a Faster R-CNN, is above a threshold, and the second uses two-stream ConvNets fed with the RGBD images. In the recognition module, Depth Dynamic Images (DDIs) and Depth Motion Dynamic Images (DMDIs), are generated from a depth sequence through bidirectional weighted rank pooling and are then fed into ConvNets for classification. Saliency sequences for depth and RGB sequences are also estimated using the algorithm described in Achanta et al. (2009) and are used as input on 3D ConvLSTM networks (Zhu et al., 2017). The normalized outputs of all four networks are fused with the average-score rule in an element-wise way to obtain the final gesture label.

Both **RNN + LSTM** and **Temp Conv + LSTM Cells** are described in Pigou et al. (2018). In **RNN + LSTM**, they leverage the idea that a gesture becomes recognizable only after a few time steps, so they developed a method that uses a bidirectional RNN, which enables the processing of the sequences (frame by frame) in both temporal directions, with LSTM cells learning dynamic temporal dependencies. **Temp Conv + LSTM Cells** is an extension of CNN layers with temporal convolutions, capable of extracting hierarchies of motion features and capturing time-related information from the input videos. The proposed method's 3D convolution is factorized into two-dimensional spatial convolutions and one-dimensional temporal convolutions to avoid model overfitting. The resulting classification module is adapted to continuous detection using a sliding window with single-frame steps.

Fig. 8. Left: depth maps from the ConGD dataset. Right: the corresponding RGB images. *Source:* From http://www.cbsr.ia.ac.cn/users/jwan/database/congd.html.

Table 7
Performances of the best methods on the ConGD benchmark.

Method	Date	JI (%)
TD-Res3D + Average fusion	2018	71.63
ConvNetsFusion	2021	69.04
ST-GAT	2022	65.78

4.3. ConGD

Wan et al. (2016) created the Continuous Gesture Recognition Dataset (ConGD) based on batch classes and samples from the ChaLearn gesture dataset (CGD) (Guyon et al., 2012). The CGD dataset was designed for a series of "One-shot learning" tasks, where there is only one training example for each gesture, and the rest of the data is used for testing. The tasks correspond to different application domains, e.g., sign language, helicopter signals, traffic signals, and underwater sign language. For each of them, a specific batch of data with 100 example gestures from a small lexicon of 8 to 12 classes has been created. The entire dataset consists of 289 classes from 30 lexicons and a total of 54,000 gestures recorded in 23,000 RGB-D videos. The ConGD dataset was obtained by annotating the videos with a semi-automatic procedure, combining the different batches, removing some of them, and fusing the classes with similar characteristics. The final benchmark features a dictionary with 249 gesture classes, and it is composed of 22,535 RGB video recordings and 22,535 depth videos where 21 subjects perform them for a total of 47,933 executions (Fig. 8). The dataset is split into a training set with 30,442 gesture samples, a validation set with 8889, and a test set with 8602. The Jaccard Index is the only metric used to measure the continuous recognition performance on this benchmark.

4.3.1. Best results on ConGD

Table 7 shows the best results obtained on the ConGD benchmark.

Temporal Dilated Residual ConvNet (TD-Res3D + Average fusion) is an approach proposed by Zhu et al. (2019). This procedure begins with segmenting the continuous gesture sequences into isolated gesture instances with a temporal-dilated Res3D network (Tran et al., 2017). The authors employed a balanced squared hinge loss function to handle the imbalance between boundaries and non-boundaries. The temporal dilation preserves temporal information for precise boundary detection, while a large temporal receptive field enhances the accuracy and effectiveness of the segmentation results. The method uses this segmentation network on both RGB and depth streams. The results are then merged with the average fusion score. The classification network is a combination of a 3D Convolutional Neural Network (3DCNN), a Convolutional Long-Short-Term-Memory Network (ConvLSTM), and a 2D Convolutional Neural Network (2DCNN) for isolated gesture recognition. This architecture is adapted to learn long-term and deep spatio-temporal features, making it effective to recognize the gestures of this benchmark.

The **ConvNetsFusion** method (Wang, 2021), already described among the best methods for Montalbano Track 3, presents good results on the ConGD benchmark as well.

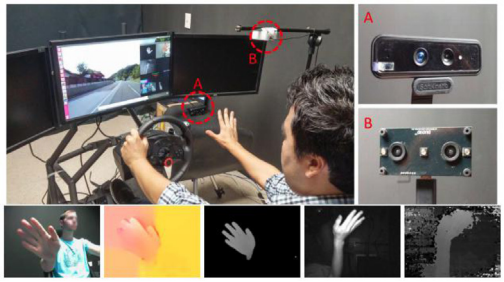

Fig. 9. Top left: recording setup for the NVGesture dataset. The subject uses his left hand to drive and performs the gestures with his right hand only. NVGesture includes multi-modal acquisitions. Data are captured with a SoftKinetic DS325 recording front-view RGB-D videos (A), and with a DUO 3D stereo-IR camera mounted above the user (B).
Source: From https://www.v7labs.com/open-datasets/nvgesture.

Table 8
Performances of the best methods on the NVGesture benchmark. (*) metric from Köpüklü et al. (2020).

Method	Date	τ	AUC	TPR (%)	FPR (%)	NTtD	LA (%)*
R3DCNN	2015	0.3	0.93	88	15	0.56	NA
HSTV	2022	0.16	0.81	85	17	0.2158	NA
ResNet+ResNeXt	2020	0.15	NA	NA	NA	NA	77

Spatial–Temporal Graph Attention Network (ST-GAT) (Guo et al., 2023) focuses on extracting essential features from video sequences by considering local details of the hand movements. The approach takes joint and bone information as inputs and constructs a spatial–temporal graph to capture inter-frame relevance and physical connections between nodes. It then exploits a graph-based multi-head attention mechanism with adjacent matrix calculation to effectively explore the local features. Additionally, it models the short-term motion correlation with a temporal convolutional network and uses a bidirectional long short-term memory (BLSTM) to capture long-term dependencies in the video sequences. Finally, it applies frame-by-frame the connectionist temporal classification to align the word-level sequences.

4.4. NVGesture

The NVGesture dataset (Molchanov et al., 2016), is designed to test gesture recognition for touchless driver control. The authors recorded it using multiple sensors and viewpoints; this involved 20 subjects who performed the gestures with their right hands while holding the steering wheel with their left hand. The SoftKinetic DS325 sensor was used to acquire front-view color and depth videos, and a top-mounted DUO 3D sensor was used to collect this dataset for recording a pair of stereo-IR streams (Fig. 9). Data are captured at a frame rate of 30 fps.

The dictionary includes 25 gesture types, and the benchmark contains 1532 dynamic hand gesture samples; these samples are divided into 1050 training and 482 test samples. The metrics chosen for the evaluation are the area under the curve (AUC) of the receiver operating characteristic (ROC) curve and the normalized time to detect (NTtD) at a detection threshold ($\tau = 0.3$).

4.4.1. Best results on NVGesture

Table 8 shows the best results obtained on this benchmark. **Recurrent 3D convolutional neural network (R3DCNN)** was proposed in the paper that introduced the benchmark (Molchanov et al., 2015). The method splits the entire video into fixed-length clips, computes the class conditional probabilities set for each clip, and averages them across the modalities. The architecture includes a deep 3D-CNN for spatio-temporal feature extraction, a recurrent layer for global temporal modeling, and a softmax layer for predicting the class-conditional

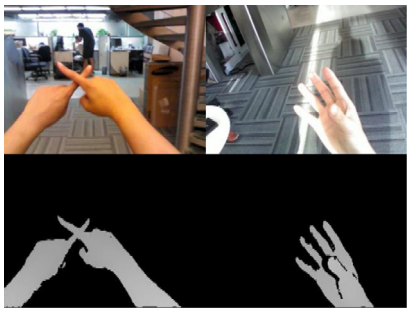

Fig. 10. Example RGB (top row) and depth images (bottom row) from the EgoGesture dataset.
Source: From http://www.nlpr.ia.ac.cn/iva/yfzhang/datasets/egogesture.html.

gesture probabilities. The authors also tested a support vector machine (SVM) instead of the softmax layer to compute the final classification score.

HandShapeTemporalVariation (HSTV) was proposed in Wannous and Vandeborre (2022). It is a deep learning-based approach that effectively utilizes the combined description of the hand shape and its temporal variation. To achieve this, they trained a CNN (Tang et al., 2014) on a depth image dataset for hand pose estimation. The network creates two distinct hand representations for each time step: fine features, representing the pose, and coarse features, roughly representing the hand shape. The features are fed into two different Recurrent Neural Networks (RNNs), RNNfine and RNNcoarse, modeling pose and shape variations over time. RNNs' outputs are merged with a joint-fine-tuning method that retrains the two last softmax layers while forcing their sum to represent both networks. The online classification is handled by training the recurrent network with the gesture dictionary and a non-gesture class. The algorithm returns at each frame a class probability. A threshold for considering a gesture classification valid is obtained from the training data with a heuristic trying to maximize true positives and minimize the false positive rates.

ResNet+ResNeXt (Köpüklü et al., 2020) is a two-modules algorithm. The detector is a ResNet-10 trained to distinguish gestures from non-gestures. It activates the classifier model when a gesture is detected. The classifier is a ResNeXt-101 (Xie et al., 2017), chosen after testing several resource-efficient 3DCNNs (Tran et al., 2014). The network is fed with the frames in the classifier queue sent from the detector. To prevent misclassification, the authors placed the raw softmax probability of the last detector predictions into a queue to obtain the final detector decisions with a filtering step. This method is not comparable with the others because its performances are evaluated using Levenshtein Accuracy (see Section 3.4), which is not used in the benchmark description (Molchanov et al., 2015) and in other works (reported in column LA of Table 8).

4.5. EgoGesture

The EgoGesture dataset (Zhang et al., 2018), includes egocentric RGB-D videos of hand gestures and is specifically designed to evaluate methods to interact with wearable devices like VR and AR headsets. Videos in this benchmark were captured using a head-mounted Intel RealSense SR300 RGB-D camera with a 640 × 480 resolution and a frame rate of 30 fps (Fig. 10). The dictionary comprises 83 static or dynamic gesture classes. The authors divided these classes into two types: Communicative and Manipulative. Communicative gestures represent choices or commands for interfaces (numbers, symbols). Manipulative gestures are designed to control actions on interface components, such as zoom, rotate, and open/close. The dataset includes 300 samples for each class with a large intra-class variety. Gesture samples are included in 24,161 RGB-D video sequences and are performed by 50 subjects. The video recordings have strong variations in background and illumination conditions and have been captured in indoor and outdoor environments. The start and end frame indexes of each sample are manually labeled. The minimum length of a gesture is three frames, while the maximum is 196 frames. The sequences were randomly split into a training set (1239), a validation set (411), and a test set (431). The evaluation metrics used for this benchmark are the Jaccard Index and the execution runtime of the tested methods measured in frames per second (fps).

Table 9
Performances of the best methods on the EgoGesture benchmark. (*) metric from Köpüklü et al. (2020).

Method	Date	JI (%)	Runtime (FPS)	LA (%)*
TMMF	2021	80.3	NA	NA
C3D	2015	71.8	112	NA
ResNet+ResNeXt	2020	NA	NA	91

4.5.1. Best results on EgoGesture

Table 9 shows the best results obtained on the EgoGesture benchmark. In the following, we report a description of the corresponding methods.

Temporal Multi-Modal Fusion (TMMF) (Gammulle et al., 2021) is a sliding window method that does not require a preliminary segmentation step. Deep features are extracted from windows that slide along the video. Data are processed by individual Uni-modal Feature Mapping (UFM) blocks, composed of temporal convolution layers and multiple dilated residual blocks. The output feature vectors are passed through the fusion block, which selects and extracts temporal features using the attention level parameter to create the fused feature vector. This discriminative feature vector is the input to the Multi-modal Feature Mapping (MFM) block that performs the classification. The multi-modal fusion mechanism is scalable to any number of modes. **3-dimensional convolutional networks (C3D)** (Tran et al., 2015) also uses a sliding-window approach. C3D uses spatio-temporal convolutional layers to extract features from a fixed-length window sliding along the video stream. In detail, the network is composed of eight 3D convolutional layers, one 2D pooling layer, four 3D pooling layers, three fully connected layers, and, finally, an LSTM layer. This method was tested on EgoGesture by the benchmark's creators and obtained the best results among the algorithms tested in the original paper (Zhang et al., 2018). **ResNet+ResNeXt** is the same method already presented in Section 4.4.1 as tested on NVGesture. It has also been tested on EgoGesture and evaluated using the LA event-based evaluation metric, different from those used in the original paper. This method is not comparable with the others since the metrics are different.

4.6. SHREC'19

The organizers of the SHREC 2019 Track: Online Gesture (Caputo et al., 2019) created this benchmark, by capturing the hand movements of 13 subjects while interacting with a 3D interface in an immersive VR environment. The subjects wore a Head Mounted Display (Oculus Rift) with a Leap Motion sensor placed on it to capture hand poses from an egocentric point of view. The dictionary is composed of 5 different coarse gestures characterized by a 2D hand trajectory (Fig. 11). The dataset includes 195 hand pose sequences, each containing a single gesture in a different time location. The non-gesture part of the sequences is roughly determined by a set of specific actions on the virtual interfaces required by the interactive VR app. For the training set, 60 of these sequences with manually annotated start/end of the gestures are used, and the rest of the data are used as the test set. In the evaluation, a gesture is considered correctly detected and labeled if the recognition method locates it within 2.5 s from the actual gesture time window in the sequence. Given the detected and ground

Table 10
Performances of the best methods tested on the SHREC'19 benchmark. (*) metric from Cunico et al. (2023). CD: percentage of gestures detected in the correct time window and correctly classified, ML: gestures detected in the correct time window and mislabeled, FP: percentage of false positives encountered before the true gesture, M: percentage of missed detections, T1 Err.: average error in estimating the initial mark T1, T2-e: average distance, in time, of the detection frame from the end mark, T2-s: average distance of the detection frame from the start mark.

Method	Date	CD	ML	FP	M	T1 Err. (s)	T2-e (s)	T2-s (s)	Time (ms)*	FPR*
OO-dMVMT	2023	88	NA	NA	NA	NA	NA	NA	5.8	5
uDeepGRU	2019	85.2	7.4	3	4.4	0.54	−1.66	0.66	3.0	10
SW-3 cent	2019	75.6	16.3	2.2	5.9	0.58	−0.7	1.61	1.7	NA

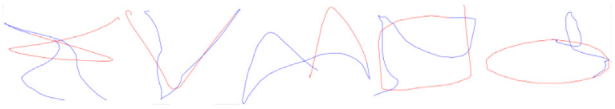

Fig. 11. The dictionary of the SHREC'19 benchmark features 5 gesture types characterized by simple 2D paths (red lines) hidden in longer 3D non-gesture trajectories (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
Source: From Caputo et al. (2019).

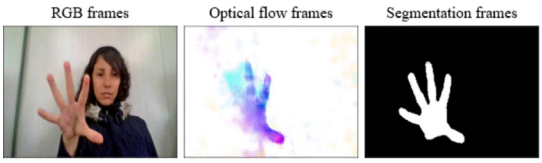

Fig. 12. The different data types included in the IPN Hand distribution. From left: RGB video frames, optical flow maps and segmentation masks. From https://gibranbenitez.github.io/IPN_Hand.

truth gesture, the following metrics are estimated to characterize the methods' performances: the percentage of correct detections, timely detections with wrong classification label, detections made before the correct time location, detections made after the correct time window or missed, the time difference between the decision time of the online algorithm and the actual start and end of the gestures. In the Cunico et al. (2023), the benchmark is used by adding the classification time and False Positive Ratio metrics.

4.6.1. Best results on SHREC'19

Different methods have been applied to the SHREC'19 benchmark, and the best results are summarized in Table 10.

On-Off deep Multi-View Multi-Task (OO-dMVMT) (Cunico et al., 2023) is based on a fixed-size sliding window approach. This method exploits the multi-view classification paradigm, where multiple complementary features derived from the data in the input window are used to create a robust embedding. These features/views are derived from the DD-Net method (Yang et al., 2019) and are the Joint Collection of Distances (JCD), representing the hand pose, and the Motion-Fast and the Motion-Slow vectors representing the short-term global motion of the skeleton in terms of speed. OO-dMVMT also exploits the multi-task (MT) paradigm, as in the sliding window training framework, the optimization is based on 4 tasks: (1) gestures classification into three rough categories, Static, Dynamic, and Non-Gestures; (2) gesture classification with the fine-grained label; (3 and 4) detection of gestures' start and end. The on-off mechanism enables/disables the heads related to tasks 3 and 4 when the training window includes or does not include annotations of gesture limits.

uDeepGRU, by Maghoumi et al. is an extension of the Deep GRU approach proposed by the same authors (Maghoumi and LaViola, 2019), used by them to participate in the SHREC'19 contest (Caputo et al., 2019). It adapts the original method to work in online application scenarios. uDeepGRU is an end-to-end deep learning-based unsegmented gesture recognizer. It features an encoder and a classification sub-network. The former uses Gated recurrent units (GRU), and the latter exploits fully connected layers followed by batch normalization and dropout. To reduce the over-fitting due to the small size of the SHREC'19 training set, the authors removed the attention sub-network used in the original DeepGRU method. To cope with the limited amount of training data, this method also puts a particular emphasis on data augmentation, using four different methods: stochastic resampling (GPSR), Fourier coefficient perturbations, time-series inversion, and rotation. This method obtained the best performances in the SHREC'19 contest.

SW-3cent was the baseline method proposed in Caputo et al. (2019), derived from a simple trajectory comparison-based technique called 3-cent (Caputo et al., 2018) and applied in a sliding window approach. The method tests sliding windows of different sizes (the average size of the gestures in the training classes), comparing them with the labeled examples. The comparison involves the resampling of the trajectories to obtain the same number of points using spline interpolation, the application of transforms to superimpose centroids, and scaling to fit a fixed-size bounding box. Finally, the trajectory distances are estimated as the sum of squared distances between pairs of corresponding points.

4.7. IPN Hand

This dataset (Benitez-Garcia et al., 2021b), focuses on interaction with touchless screens. It contains 100 RGB video recordings of sequences, including 4218 gesture samples, performed by 50 different individuals. The videos were captured with PC or laptop cameras, keeping a fixed resolution (640 × 480) and frame rate (30 fps). Authors provide not only video data and annotations but also optical flow sequences and accurate hand segmentation masks estimated as described in Benitez-Garcia et al. (2021b) (Fig. 12). The sequences are captured with 28 different background environments, both static and dynamic, with strong variations in background clutter and lighting conditions (both strong and weak illumination). Each video's start and end frame index for gesture instances was manually labeled. The minimum length of a gesture is nine frames, and the maximum is 650 frames. The dataset is split into a training set with 148 videos comprising 3117 gesture instances performed by 37 subjects and a test set with 52 videos and 1101 gesture instances performed by 13 subjects. The 13 gesture types are designed for desktop-like interface control: two static pointing gestures (with one or two fingers) and 11 dynamic gestures (clicking with one and two fingers, throwing to four positions, double-clicking with one and two fingers, zoom-in, and zoom-out). The metrics used for the evaluation are the Levenshtein accuracy, the model size, and the inference time.

4.7.1. Best results on IPN Hand

Table 11 shows the best results obtained on the IPN Hand benchmark.

EUREKA (Peral et al., 2022) consists of two main modules: the first focuses on the localization of landmarks in 2D images, and the second predicts the hand gesture class. In this architecture, the Google Mediapipe (Zhang et al., 2020) detector is applied to find the hand landmarks. A scaling- and translation-invariant feature vector is then extracted from the raw landmark positions and used as the input for a densely connected convolutional neural network to classify hand

Table 11
Performances of the best methods tested on the IPN Hand benchmark.

Method	Date	LA (%)	Mod.size (MB)	Infer.time (ms)
EUREKA	2022	87.5	NA	NA
FASSD-Net	2021	77.26	NA	28
TMMF	2021	68.12	NA	NA

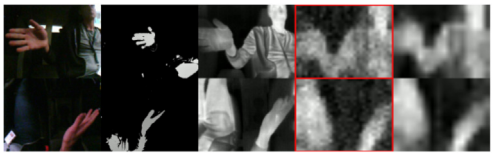

Fig. 13. Sample RGB and thermal frames from the front sensor cluster (top row) and top sensor cluster (bottom row) included in the Mlgesture dataset.
Source: From Vandersteegen et al. (2020).

gestures. To recognize dynamic gestures, encoding the spatial changes among landmarks' positions and over time, they proposed a frame selection strategy. For each gesture, they selected 15 keyframes from which they extracted hand positions and three features (Distances, DistAndTime, DistTime) to be used as inputs to the neural network. The keyframe selection is not trivial in continuous time, which is why a multiple-sized window approach is executed. This approach consists of taking the model's prediction result of three different-sized windows and giving priority to the window with the highest score.

Fast and Accurate Semantic Segmentation with Dilated Asymmetric Convolutions (FASSD-Net) (Benitez-Garcia et al., 2021a) employs a lightweight semantic segmentation approach to boost the accuracy of two efficient hand gesture recognition methods: the Temporal Segment Networks (TSN) and the Temporal Shift Modules (TSM). The FASSD-Net is based on the Harmonic Dense-Net architecture (HarDNet Chao et al., 2019) for the real-time semantic pixel segmentation (this architecture can segment pixels into the background, human shape, left and right hands). Two main modules are added to the U-shape encoder–decoder of HarDNet to increase the segmentation performance. Dilated Asymmetric Pyramidal Fusion (DAPF) increases the encoder's receptive field, while Multi-resolution Dilated Asymmetric (MDA) fuses and refines multi-scale feature maps that are deeper stages of the network's outputs. To divide the sequences into clips, the authors took their inspiration from the Temporal Segment Network (TSN) (Wang et al., 2016); this model presents a two-stream CNN combining RGB-based and Optical Flow-based networks. However, in this work, a single CNN is used (RGB input). Since the model learns frame-wise, it cannot infer any temporal relationship, so the authors added a temporal shift module (TSM, Lin et al., 2019) to shift part of the channels along the temporal dimension, facilitating information exchanged among neighboring frames.

Temporal Multi-Modal Fusion (TMMF) (Gammulle et al., 2021) was already introduced for the EgoGesture dataset (Section 4.5.1); this method has also been applied to the IPN-Hand benchmark, but the results are significantly worse than those obtained using the best method.

4.8. MlGesture

This benchmark (Vandersteegen et al., 2020) is based on the first publicly available hand-gesture recognition dataset based on low-cost thermal sensors (Melexis MLX90640/MLX90641) data. It aims at evaluating methods to control a multimedia system in a car. To investigate the impact of sensor type and viewpoint, two sensor clusters were used to capture data, each consisting of 5 different devices (the low-cost MLX90640 and MLX90641 thermal cameras, an OpenMV color cam, an MLX75027 time-of-flight depth sensor, a FLIR lepton thermal camera).

Table 12
Performances of the best methods on the Mlgesture benchmark for the low-latency continuous detection task.

Method	Date	mAP (1-frame latency)
MC TCN	2020	0.83
BI-LSTM	2020	0.74

One sensor cluster was mounted on the center of the car's dashboard in front of the driver, while the other was mounted on the ceiling, pointing straight down (Fig. 13) . The dictionary consists of 9 different dynamic hand gestures compatible with low and ultra-low-resolution sensors. The authors acquired over 1300 RGB, RGB-D, and thermal video clips for both training and evaluation purposes, but only the thermal data are released in the public dataset. 24 different subjects performed the gestures in the videos. In addition to the gesture videos, non-gesture videos were also recorded, capturing actions like steering, gear switching, operating the radio, and using the wipers. Tests for continuous gesture detection were performed on a sequence obtained by stitching half of the test gesture and non-gesture acquisitions in a single long video and manually annotating the hand gesture nuclei in the resulting test video. In the continuous evaluation, if a correct prediction is within an annotated gesture nucleus, it is considered a True Positive, while further detections during the nucleus or detections outside the nucleus are considered False Positives. Missed gestures are considered False Negatives. In this benchmark, the metric used for the comparisons is the Mean Average Precision (plotted versus network output position graphs).

4.8.1. Best results on Mlgesture

Table 12 shows the best results obtained on the Mlgesture benchmark.

Mixed-Causal Temporal Convolution Network (MC TCN) (Vandersteegen et al., 2020) use a sliding window approach for the online recognition. The window classifier exploits a 2D CNN to generate an embedding vector for each video frame. The temporal domain is then modeled using a 1D TCN. The fundamental block of the architecture, called BB, is composed by multiple TCN basic blocks. The first layer of BB consists of a 1D dilated convolution with a kernel size of $k = 3$. The dilation factor is doubled for each subsequent BB. A group of consecutive BBs forms a stage. To create deeper networks, multiple stages can be stacked on top of each other, and the dilation factor is reset to 1 at the beginning of each new stage. This design enables the network to capture temporal dependencies effectively by increasing the receptive field through dilated convolutions in the TCN, facilitating better recognition and understanding of complex spatio-temporal patterns in video data. To boost the performance for predictions close to the right edge, the method uses causal convolution combined with a regular (non-causal) dilated convolution. For this reason, the configuration is called Mixed-Causal. The authors also proposed a variant of the architecture employing the SqueezeNet V1.1 (Iandola et al., 2016) spatial encoder instead of the ResNet18, resulting in a large reduction in the number of parameters with limited loss in accuracy.

Bidirectional LSTM Networks (BI-LSTM) The previous method is the top performing for the continuous task in the paper presenting this benchmark (Vandersteegen et al., 2020). Among the other methods compared in the same paper, the second best was Bidirectional LSTM Networks (BI-LSTM,Rocha and Cardoso, 2004). In the bidirectional approach, inputs are processed in two directions: from past to future and from future to past. This bidirectional approach provides a more comprehensive understanding of the sequence and contributes to the model's enhanced performance. Vandersteegen et al. (2020) used BI-LSTM with the same sliding window protocol adopted for MC TCN and other classifiers for the online recognition.

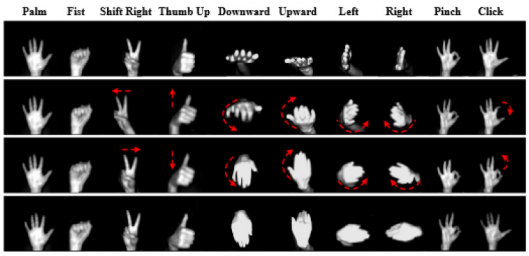

Fig. 14. The gesture dictionary of the LD-ConGR dataset.
Source: From Liu et al. (2022).

Table 13
Performances of the best method tested on the LD-ConGR benchmark.

Method	Date	JI (%)
ResNeXt-MMTM	2022	34

4.9. LD-ConGR

The Large RGB-D Video Dataset for Long-Distance Continuous Gesture Recognition (Liu et al., 2022), is characterized by a front view of the subject ranging from 1 to 4 meters away. The videos are captured with a Kinect V4 device at a resolution of 1280 × 720 for the color streams and 640 × 576 for the depth streams, with a frame rate of 30 fps. The dictionary includes 10 fine-grained hand gesture classes (Fig. 14), 3 of which are static (palm, fist, thumb up), and 7 are dynamic (shift right, downward, upward, left, right, pinch, click). The authors collected 542 videos from 30 subjects in 5 meeting rooms (with different designs and furnishing); there are 6 recording spots in every meeting room. Each video contains multiple gestures, and all the gesture instances were manually annotated with the label and the starting and ending frames. Examples of correct gesture executions were shown to the subjects before the recording started. The gestures were performed continuously, allowing short breaks between gesture instances. Given the large distance, the hands are small and difficult to recognize, and the authors added their position as an annotation for each frame. There are a total of 44,887 gesture instances in the videos. These videos are divided into training and testing sets, with 34,315 and 10,572 gestures (performed by 23 and 7 subjects), respectively. The evaluation of the continuous recognition is done with the Jaccard index.

4.9.1. Best results on LD-ConGR

Table 13 shows the best result obtained on the LD-ConGR benchmark.

Resnet-Xt Multimodal Transfer Modules (ResNeXt-MMTM) is the model proposed for this benchmark in Liu et al. (2022). The authors built a baseline method using ResNeXt-101 (Xie et al., 2017) and tested the recognition performances on single (RGB or depth) input modalities. Additionally, they developed a multimodal fusion model ResNeXt-MMTM inspired by Joze et al. (2019), combining the features of different modalities at multiple layers through Multimodal Transfer Modules (MMTMs). MMTM learns a multimodal embedding and fine-tuning the features of each modality. In their architecture, features extracted from both the RGB and depth streams are combined by the MMTM after each ResNeXt block. After passing the fully connected layers, the feature vectors are integrated with an element-wise addition and passed through a softmax layer to have the prediction of the result. For the continuous recognition test, the classifiers are used on a 32-frame window sliding over the video sequence (with a temporal stride of 2 frames).

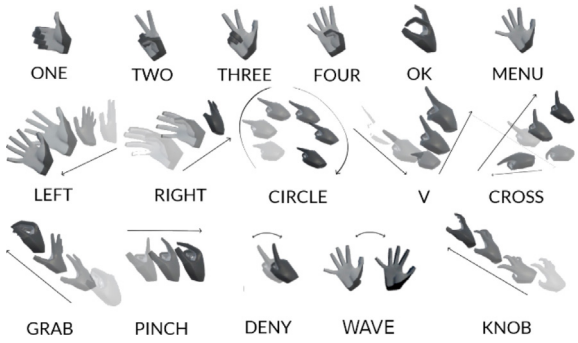

Fig. 15. The gesture dictionary of SHREC'22 includes static (top row), dynamic coarse (middle row), and dynamic fine and periodic gestures (bottom row).
Source: From Emporio et al. (2022).

4.10. SHREC'22

The dataset of the SHREC 2022 Track on Online Detection of Heterogeneous Gestures (Emporio et al., 2022) is designed to test continuous gesture recognition for a generic mixed-reality interaction. The publicly available data provided to the contest participants are hand-pose skeleton streams directly estimated by the finger-tracking system integrated into the Hololens 2 headset. In the tracker's data, each frame contains the coordinates of 26 right-hand joints with a relatively low and not perfectly stable frame rate (approximately 20 fps). The dictionary features 16 heterogeneous gestures, including static (a fixed hand pose for at least 0.5 s), dynamic-coarse (characterized by whole hand trajectory), dynamic-fine (where the semantic also depends on finger articulations), and periodic (with repeated movements) gestures (Fig. 15). The dataset is divided into a training and a test set, with the same number of sequences (144) and instances (36) of each gesture class. Each sequence has a minimum of 3 and a maximum of 5 gestures. Two different groups, each with three subjects, recorded the training and testing sequences. The training set is annotated with each gesture's label, start frame, and end frame. The evaluation is based on a set of different metrics: detection rate, false positive ratio, Jaccard Index, classification time, average temporal delay, and the average delays in frames of the predicted gesture.

4.10.1. Best results on SHREC'22

Table 14 shows the best results obtained on the SHREC'22 benchmark. In the following, we report a description of the corresponding methods.

On-Off deep Multi-view Multi-Task (OO-dMVMT), already described in Section 4.6.1 is particularly effective for the SHREC'22 benchmark. In this case, the task is more challenging due to the heterogeneous gesture dictionary. The authors (Cunico et al., 2023) used the same features for the recognition and adapted the sliding window size for the detection depending on the dataset.

Two-Model-Based Online Hand Gesture Recognition (Two-Model), proposed in Doždor et al. (2023) is an online gesture detection system based on two models. The first model (gesture localizer) is a binary classifier that can identify a gesture's execution. Segments that contain gestures are resampled to a fixed length and provided as input to the second model (gesture recognizer), which classifies them into one of the known gesture classes or the non-gesture class. For each input window, the per-axis coordinates are normalized to obtain zero mean and unit variance; joint velocity and pairwise Manhattan distances are derived from the coordinates, flattened, and concatenated to the input to obtain the input feature vector. The encoder's architecture consists of a fully connected layer followed by two GRUs layers with hyperbolic tangent activation. The classification model consists of only one fully connected layer with a sigmoid activation function.

Table 14
Performances of the best methods tested on the SHREC'22 benchmark. DR: Detection Rate (%), FPR: False Positive's Ratio (%), JI: Jaccard Index (%), D-End/D-Start: average delays in frames from the ground truth end/start of the annotated gestures (depending on the algorithms' design).

Method	Date	DR	FPR	JI	Delay (fr.)	Time (ms)	D-Start (s)	D-End (s)
OO-dMVMT	2023	92	9	85	8	4	NA	NA
Two-Model	2023	85	9	78	5	20	NA	NA
Causal TCN	2022	80	29	68	19	28	4.36	−28.79

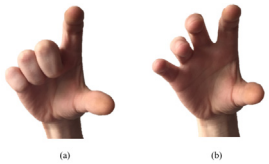

Fig. 16. The different ways to perform gestures in the OHDG (and DHG 14/28) dataset: using (a) one finger or (b) the whole hand (from http://www-rech.telecom-lille.fr/DHGdataset/).

Causal temporal convolutional network (Causal TCN) obtained the best results in the SHREC'22 original contest. The method, proposed by Ambellan et al. and described in Emporio et al. (2022), is based on a temporal convolutional network (TCN) employing causal filters to create a lightweight classification structure. The authors implemented a sliding window approach for short-time windows to reduce the complexity of the learning task. In particular, each window is fed through two convolutional layers composed of causal or unidirectional convolutional filters, and a fully connected linear layer gives the classification result. A per-frame labeling is obtained from the sliding window classification in the testing phase using a voting procedure, and a post-processing step is performed to derive the gestures' segmentation.

4.11. ODHG 14/28

The Online Dynamic Hand Gesture dataset (Wannous and Vandeborre, 2022) is derived from the well-known segmented gesture recognition benchmark called DHG-14/28 (De Smedt et al., 2016). This dataset was intended to be a segmented gesture benchmark, even though it was captured recording continuous sequences with both gestures and non-gesture parts. Data consists of RGB-D sequences captured with an Intel RealSense camera and the corresponding hand skeleton sequences extracted with the RealSense API. The depth image resolution is 640 × 480, and both image and hand skeletons are recorded at 30 fps. The dictionary includes 14 gesture classes: five "dynamic fine" (Grab, Expand, Pinch, Rotation clockwise and counterclockwise), characterized by finger articulation, and nine "dynamic coarse" (Tap, Swipe Right, Swipe Left, Swipe UP, Swipe Down, Swipe X, Swipe V, Swipe +, and Shake), characterized by global hand movements. The gestures have variable execution times, ranging from 20 to 50 frames. There are two gesture execution methods: using only two fingers or using the whole hand (Fig. 16). If the same gesture class is assigned to both methods, there would be 14 gesture labels; otherwise, there would be 28 gesture labels. The whole dataset comprises 280 sequences performed by 20 subjects and ten gestures each. Each subject executed each gesture five times. While the original benchmark only tests the classification of segmented gestures from the captured data, Wannous and Vandeborre (2022) reused the sequences to create a continuous gesture detection benchmark, the Online DHG (ODHG). In this version, a class label is annotated for each frame of the sequences. The frames that occur between meaningful gestures are labeled as the "no gesture" class. During these intervals, the participants were allowed to take a resting position without any specific instruction. To evaluate the online detection and recognition, different metrics are used: accuracy, Receiver Operating Characteristic (ROC), False Positive Rate (FPR), True Positive Rate (TPR), and Normalized Time to Detect (NTtD).

Table 15
Performances of the best methods on the ODHG benchmark.

Method	Date	AUC (%)	TPR (%)	FPR (%)	NTtD
HSTV	2022	91	85	17	0.2104

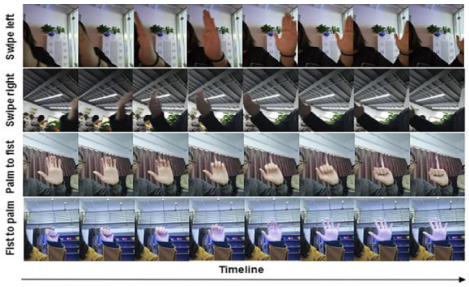

Fig. 17. Examples of gestures sequences included in the ZJUGesture dataset, captured by different mobile devices.
Source: From Xu et al. (2023).

4.11.1. Best results on ODHG

Table 15 shows the best results obtained on the ODHG benchmark.

HandShapeTemporalVariation (HSTV) proposed in Wannous and Vandeborre (2022) and already described in Section 4.4.1 was applied to this benchmark by their creators. Also in this case the authors tuned a probability detection threshold based on the ROC curve plot, obtaining the results plotted in Table 15. The false detection rate is clearly too high for a practical use of the system.

4.12. ZJUGesture

This dataset was created by Xu et al. in 2022 (Xu et al., 2023), with the primary focus on one-handed operation scenarios commonly encountered in mobile phones or tablets, and includes RGB video sequences recorded with these devices (Fig. 17). The benchmark's dictionary includes nine categories of gestures designed to be user-friendly and easily operable. Each gesture is divided into preparation, core action, and retraction. To ensure the diversity of the captured samples, each gesture is recorded in 12 sub-scenes, which include various backgrounds and different light intensities. This approach helps to capture variations that may occur in real-world usage scenarios. All videos are labeled frame by frame with the gesture's name; each video has a resolution of 1280 × 720 pixels, captured at a frame rate of 30 FPS. In the ZJUGesture, 60 people performed each gesture according to their habits at three speeds: fast, normal, and slow. The Levenshtein Accuracy is used to evaluate the online task together with three efficiency measures: speed, GFLOPs, and number of parameters.

4.12.1. Best results on ZJUGesture

Table 16 shows the best results obtained on the ZJUGesture benchmark. It is necessary to note that since the ZJUGesture benchmark was introduced in 2022, the only available results are those presented in the original paper.

MotionNet + GestureNet, is an online lightweight two-stage framework for the detection and classification of dynamic gestures proposed

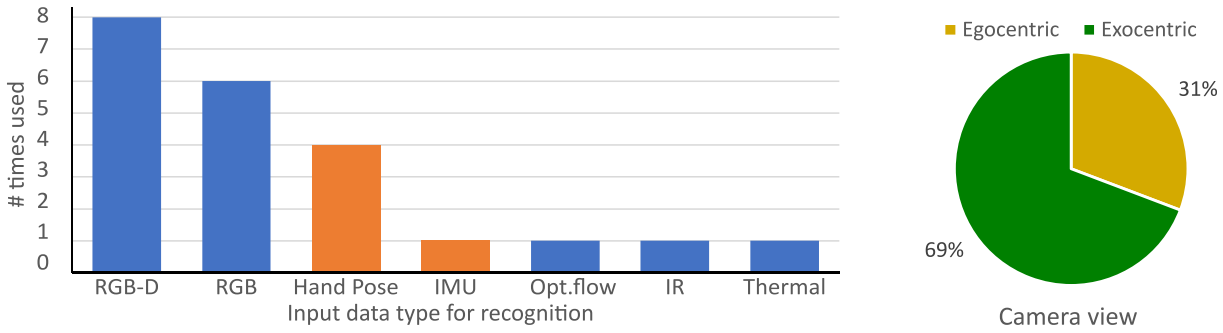

Fig. 18. (a) Data types used as input for the recognizers in the different benchmarks. Blue bars indicate video streams, and orange bars key point data streams. Most of the benchmarks are based on videos. (b) Most of the benchmarks feature data acquired from an exocentric view. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 16
Performances of the best methods on the ZJUGesture benchmark.

Method	Date	LA (%)	Speed (seg/s)	GFLOPs	Param. (M)
MotionNet + GestureNet	2023	92.4	173	2.0	10
P3D ResNet	2017	78.2	57	15.9	25
C3D	2015	76.1	28	129.9	99

by the creators of the benchmark (Xu et al., 2023). The architecture is composed of a detection module and a gesture recognition module. Input videos are pre-processed by using a sliding window with a length of 4 and a stride of 2. Differential images are also estimated and concatenated to the RGB images. The detection module, MotionNet (Wu et al., 2020), aims to determine if there is a gesture execution in the raw video stream and then sends the isolated gesture to the recognition module. Their gesture classification module, GestureNet (Chang et al., 2014), is a temporal relation reasoning network that aims to process the cleaned gesture sequences and identify them. Furthermore, they have added a state machine to process the results of the gesture recognition network so that each classified gesture responds to a single-time prediction. **Pseudo-3D Residual Net (P3D ResNet)**, proposed in Qiu et al. (2017), is a network architecture that leverages all the variations of blocks found in ResNet (He et al., 2016) but arranges them in different positions within the architecture to enhance the structural diversity of the network. Pseudo 3D CNN reduces the model size of a ResNet and enables the pre-training of 2D CNN from image data, increasing the leveraging power of the knowledge of scenes and objects learned from images. The original method only supported the trimmed video clips containing a single gesture, so the ZJUGesture's authors used their detection module to cut the video into small sequences and pass them through the P3D ResNet.

3-dimensional convolutional networks (C3D), already described in Section 4.5.1, was also tested on the ZJUGesture benchmark (Xu et al., 2023). The Levenshtein accuracy is lower than those obtained with the previous techniques, but the method is significantly faster than the others.

5. Discussion

The outcomes of our survey are pretty interesting and show that while gesture recognition is a hot topic in the literature and promising results are being obtained in the field, there is a need for new efforts aimed at creating benchmarks and evaluation methods to test recognition systems designed for interactive applications in realistic settings. In the following, we discuss the characteristics of the surveyed benchmarks and the evaluation results, pointing out the main issues found in our analysis and trying to propose new research directions for future work that take inspiration from them.

5.1. Benchmarks' features

5.1.1. Few benchmarks are available

Despite our extensive search, we did not find many benchmarks, and it is clear that more effort is needed to allow the scientific community to compare different approaches for continuous gesture recognition.

The ones we found are mostly built around specific tasks (as shown in Tables 2 and 3), making it difficult to define general categories or group them meaningfully for fine-grained comparisons.

Looking at Fig. 18(a), it is evident that the data mostly comprises of RGB-D images, and there are no public benchmarks for continuous recognition based on gloves or sEMG data. There is a reasonable number of benchmarks with hand skeleton data. This feature is helpful, as XR headsets and other devices directly provide the skeleton data, resulting in a drastically simplified input for gesture recognition. However, the related datasets feature a limited number of subjects, which may be a relevant issue for generalizing their results. A further interesting aspect of the skeleton-based benchmarks, not covered in the literature, is the investigation of how much the recognition results are affected by the inaccuracy of the hand pose estimation. Skeletons are mostly obtained with computer vision algorithms applied to RGB, IR, or RGB-D images. While different finger trackers have been evaluated and compared for generic accuracy performance (Schneider et al., 2021), it is not apparent how robust gesture recognition is against pose estimation errors. The hand tracking performances may heavily depend on the position/orientation of the hand to the sensor, occlusions, and illumination; this may limit the accuracy of the recognition in practical settings, even if the systems work well when the pose estimation is correct. By looking at Fig. 18(b), it is apparent that most of the benchmarks are recorded in Exocentric view, and all of them use cameras for recording (in Ruffieux et al., 2013 IMUs are used in addition to the cameras). A more significant number of egocentric datasets with different dictionaries would be helpful to assess occlusion issues for gestures captured from this viewpoint.

A possible explanation for the limited number of benchmarks and their weaknesses is the fragmented research community working on this topic. Most of the benchmarks (8) have been proposed in conferences and journals of Computer Vision and Pattern Recognition (where algorithms' benchmarking is quite popular), others (2) have been proposed within the Computer Graphics community, and only one has been introduced in an interaction-focused journal.

One possible reason for the HCI community's limited effort is that using fixed dictionaries is a strong constraint for interaction designers, and a viable alternative would be to create a system enabling them to interactively generate and update sets of gestures to be recognized at runtime. Some recent research papers describe frameworks designed for this task (Mo et al., 2021; Shen et al., 2022; Schäfer et al., 2022). This approach follows ideas that were successfully applied to 2D gesture-based interface design.

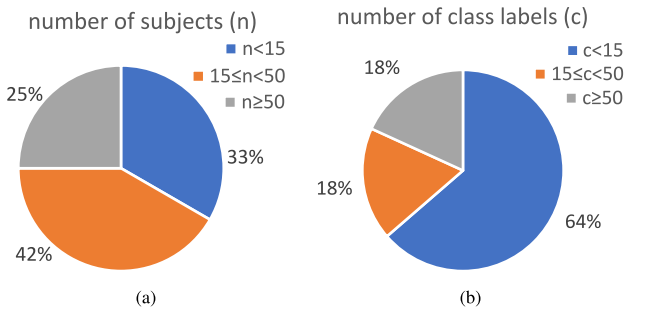

Fig. 19. (a) Most of the benchmarks feature a relatively small number of gestures classed in the dictionary (< 15), even if typically of heterogeneous types. (b) The number of subjects performing gestures is extremely variable.

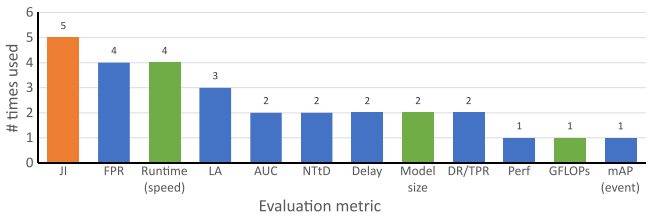

Fig. 20. Occurrence of the different evaluation metrics used to assess the method's performances in the 12 benchmarks reviewed. JI=Jaccard index, FPR=false positive ratio, AUC=area under receiver-operator characteristic curve, NTtd=normalized time-to-detect, LA=Levenshtein Accuracy, mAP=mean area precision. Orange bars indicate frame-based metrics, blue bars indicate event-based metrics, and green bars are efficiency-based metrics. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

However, testing proposed methods using realistic scenarios and standardized measurements for tasks that involve recognizing complex gestures is important. This approach helps in selecting the most effective and optimal gesture recognition methods.

5.1.2. Number and diversity of subjects

As shown in Fig. 19(a), only three surveyed datasets present a large number of subjects that executed the gestures (> 50). A limited number of subjects and the lack of precise characterization of the participants, such as age and cognitive and spatial ability, limit the generalization of the gesture recognition results to any target users.

Benchmarks' data should be captured on a large set of subjects representative of a generic population and collected with associated metadata, allowing an analysis of recognizers' performance for different subsets of target users. Gestures can be executed in different ways and with different speeds, and even the same person may change the execution due to varied physical or mental states. It is challenging to capture datasets with many subjects under different conditions, collecting all the related metadata. However, the availability of benchmarks with this variety would enormously help the development of new methods with good usability in the wild. None of the considered benchmarks evaluates the per-subject performances of the methods. This feature is essential to assess the potential gesture-based interface's accessibility regarding the recognizer. While considering bio-mechanical hand functions ergonomics (Lee and Jung, 2015), evaluating gesture dictionaries is important, especially when subjects must perform many hand poses/movements. In this situation, it is possible that the subjects cannot easily execute some of the gestures due to individual limitations. The collection of data from a large variety of subjects could also be exploited to reason about the accessibility of gestural interfaces for people with different types of abilities.

5.1.3. Analysis of gesture dictionaries

Most of the surveyed papers, including those that describe contests or introduce a benchmark, present a limited analysis of the results. In particular, a limited number of research studies have presented a per-class results analysis focusing on the effectiveness of the proposed methods for the different gesture labels. Some of the benchmarks feature a relatively high number of gesture classes (Fig. 19(b)), and evaluating how the methods perform for individual or clustered labels could provide quite interesting results.

The performance analysis on different gesture categories (static, dynamic coarse/fine) has been presented in Shen et al. (2022), and the obtained results are of interest since they show that different methods can perform better for different categories. Researchers can exploit these outcomes to design improved techniques by combining multiple recognizers. In Wan et al. (2016), the dictionary consists of related classes subsets to different application domains that enable the assessment of the performances for a particular use. However, from the interaction design point of view, an interesting research direction could be related to selecting optimal gesture subsets that are less prone to errors and more suitable for reliable user interface design. This important aspect should be considered to improve the results obtained on the benchmarks, which are far from optimal, as discussed in Section 5.2.1.

5.1.4. Metrics are not standardized

As discussed in Section 3.4, the continuous task can be evaluated with many different metrics. The surveyed benchmarks use metrics that are all different from each other. Looking at Tables 5 and 13 and Fig. 20, we see that many benchmarks evaluate continuous gesture detection using frame-based metrics only; also, the Jaccard index, which is a frame-based metric, is the most used one. We consider this a non-optimal choice, as the frame-based metrics do not test missed or repeated detection and present relevant biases in evaluating gestures of different time lengths.

We emphasize that evaluating false positives is as important as the classification accuracy for verifying the methods' suitability for practical use. Event-based metrics are also necessary since they quantify the feedback delay. However, since their values depend on thresholds (related to the detection criteria), a standardized way to define the "correct gesture detection" would be necessary. In practical use, continuous recognition algorithms run on machines with different hardware resources, and evaluating the computational requirements of the methods is also essential.

A reasonable design guideline for creating a continuous gesture recognition benchmark is to provide multiple evaluation metrics for correct event detection, false positives, detection delay, and hardware requirements. Standardizing the metrics and the gesture detection criteria would also be helpful.

The benchmarks' diversity poses a major challenge when attempting to compare results across benchmarks or to generalize evaluation methodologies and benchmark categories. A possible way forward could be to establish a set of standard evaluation metrics applicable to all benchmarks, ensuring a common ground for comparison, while allowing for the addition of task-specific metrics tailored to each benchmark's unique requirements. For instance, when deploying a recognition system on a smartwatch, memory usage (e.g., RAM consumption) becomes a critical evaluation metric that should be included alongside general ones.

5.2. Gesture spotting performances

5.2.1. Results are not really good

The best results reported in Section 4 indicate that the number of missed gestures and false positives is significant for all the benchmarks.

In many potential applications of gesture recognition, we cannot accept a detection rate of 90% or a false positive rate of 10% because,

in an extended interaction where many gestures need to be recognized, a huge amount of actions would not be activated or would be wrongly activated. Note that this issue is only captured by performing a proper event-based evaluation of the methods. In some cases, these bad results depend on the fact that benchmarks are old and not many recent techniques have been tested on them. Another reason is that few authors work on continuous tasks, whereas many papers only focus on segmented tasks. A further explanation for the sub-optimal results may be due to the challenging dictionaries, often including heterogeneous gestures with different durations and semantic characterization (static, dynamic, coarse, and fine). Specific strategies could be developed to cope with this issue, such as using different modules to detect different gesture types. Papers often report the results only as averages across all the classes in the dataset. For specific subsets of gesture labels, the accuracy in detection and the false positive ratio can be much better, meaning that it is possible to exploit these subsets to develop more reliable and usable interfaces. We did not find this kind of analysis in the reviewed works.

One of the most reasonable solutions to obtain better classification results is to improve the models' training with proper **data augmentation**. Methods to generate synthetic data to train gesture classification models have been proposed in the literature (Maghoumi et al., 2021; Shen et al., 2021; Maslych et al., 2023) and tested on segmented benchmarks. However, when it comes to data augmentation for continuous gesture recognition, there seems to be a gap. Generating synthetic examples of gestures with perturbations of the existing training data is easy. We can change the speed, position, orientation, size, or appearance of the hand, and the additional training data obtained can improve the classification results, simulating the gesture executions of different subjects. In the continuous domain, however, the real problem is different, and it is how to fill the non-gesture space with synthetic data. The characterization of sequences labeled as non-gesture and the generation of synthetic data of this kind of data is complex, as sampling the non-gesture space in the few sequences typically provided in the benchmarks is poor. The use of limited examples of non-gesture in the training phase is one reason for the non-negligible amount of false positives obtained by continuous recognizers. There is a need for new research on effective non-gesture data augmentation.

5.2.2. Different approaches

The analysis of the best-performing methods in the surveyed benchmark suggests that the two most popular approaches to address online detection, the first based on two modules (temporal detection and classification), and the second avoiding a preliminary segmentation and labeling frames or intervals with a sliding window approach, are used essentially with the same frequency. Looking at the results, there is no firm evidence of the advantages of one of the approaches, and this aspect needs further investigation. While two-module methods perform best on most of the benchmarks, in some, e.g., SHREC'22, the best method uses a single classifier model with a sliding window approach. Another relevant fact is that, in the latter work, the sliding window has been trained and is tested with fixed-length windows, meaning that the frame labeling relies on recognizing sub-parts of the gestures. This approach was successful and allows early detection with a bounded delay but could have problems in the case of dictionaries with long gestures containing similar sub-parts. In other methods, such as EUREKA (Peral et al., 2022) or SW-3cent (Caputo et al., 2019), which use both sliding windows or segmentation modules, the classifier is instead trained on complete gesture templates and tested on variable-size windows. This approach was the most successful on IPN Hand (Peral et al., 2022).

CNN is the most popular choice (Fig. 21) concerning the used classifier type for the detection/classification modules. One important reason for CNN models' popularity is that they are faster than other methods, decreasing the delay time. Using recurrent networks, transformers, or graph networks to handle skeleton data does not provide relevant advantages and may increase the complexity of the architecture. This

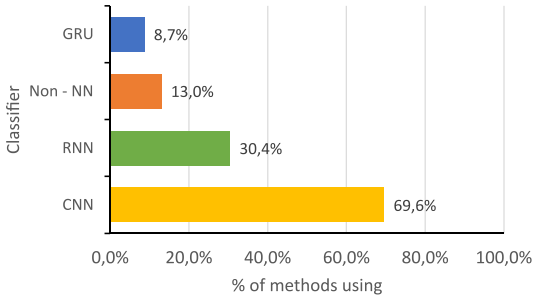

Fig. 21. Occurrence of different classifier types in the 23 methods performed applied on the proposed benchmarks and reviewed in our work.

outcome may be due to the networks' training difficulty with limited data. It is also worth noting that, even if most techniques are based on neural networks, the models' input does not always consist of the raw data. In many cases, especially for skeleton data, the input of the networks is not the original data sequence but a stream of handcrafted features derived from them (such as distances of finger key points or speeds).

5.2.3. Continuous vs. segmented task

As seen in Section 2.1, most of the methods proposed in the gesture recognition literature only focus on segmented gesture classification and are tested on related benchmarks like SHREC'17 (De Smedt et al., 2017), VIVA (Ohn-Bar and Trivedi, 2014), LAP RGB-D Isolated Gesture Dataset (IsoGD) (Wan et al., 2016), Jester (Materzynska et al., 2019). We did not consider them in our survey as we focus on continuous evaluation, but we must recognize this large amount of literature. It is evident that a large number of Computer Vision researchers prefer to develop and test methods for this simplified task, and this is also evident considering that the most cited benchmark papers analyzed here are those including both a continuous and a segmented task evaluation (NVGesture, Montalbano, Egogesture). Most works citing those papers only report tests made on the segmented task. This is probably because it is easier to train classifiers for the segmented tasks and evaluate the results, comparing them with state-of-the-art, even if we need to cope with continuous recognition in many practical applications. The outcomes of the work dedicated to the segmented task are interesting to us, as the "continuous" and "segmented" tasks are obviously linked. We can take the best methods for segmented gesture classification and use them for the continuous task by adding a sliding windows approach (Dietterich, 2002) or a gesture segmentation module. This procedure, however, is not straightforward. The training approach must be specifically designed for the continuous task, independently of the testing approach, as the non-gesture sequences interleave the gestures' data with non-meaningful hand movements that may be highly variable and present significant differences in training and test data. The specific training procedures must cope with a considerable class imbalance (non-gesture is more represented). The false positives issue needs to be addressed with care. It is impossible to understand the performances of the classifiers in a realistic continuous setting based on the classification results using the segmented benchmarks.

Adapting classifiers created for segmented tasks to a continuous setting has been proposed in Cunico et al. (2023). Using a sliding windows approach, the authors adapted the best methods on the DHG14-28 and SHREC'17 benchmarks for working online on the SHREC'22 task. They had to retrain the networks on fixed-length windows, including a non-gesture class. Other efforts of this kind could be beneficial in developing better recognizers.

5.3. Future research directions

The analysis of benchmarks and results suggests some possible research directions for future work aimed at focusing on relevant problems and improving the quality of the detection results.

- **Novel gesture recognition benchmarks are needed**, but their creation should follow specific guidelines. Large numbers of subjects should be involved, covering different user categories, and related metadata should be collected. Multimodal capture could help to analyze how much the results depend on input data acquisition and pre-processing quality. The robustness of the methods against user diversity should be tested. Basic research on algorithms should be brought closer to the problems of application domains by creating specific benchmarks for them.
- **The evaluation should be standardized** or, at least, specified for each application task in a way accepted by the scientific community. An effort to reduce the fragmentation of the community working on this topic would be desirable. We recommend the use of four primary metrics: TPR (events recognized by a method), FPR (events predicted poorly), classification time (to understand the speed of method execution), and, finally, temporal delay (to understand how fast an event is recognized). Once evaluation metrics are unified, one could compare the same method on multiple datasets by understanding how to improve it and the real difficulties of those benchmarks.
- **Optimization of gesture dictionaries** could be another goal of the evaluation. A possible idea could be to develop evaluation benchmarks/metrics for gesture dictionary usability and also for interactive gestural interface design.
- **The quality of the hand tracking** in skeleton-based classification should be assessed to separate the errors due to bad tracking from the actual gesture classification errors.
- **A careful comparison of the different continuous recognition approaches** (two modules, sliding windows trained with fixed duration subparts, sliding windows trained with segmented/resampled gestures, recurrent networks) should be performed on different kinds of datasets to understand better their advantages/disadvantages **All the methods recently proposed for the segmented task should be evaluated on continuous benchmarks**, considering the related metrics. New research should focus more on solving the issues related to training and testing for online detection rather than on trying small changes in network architectures to solve the segmented task on old benchmarks.
- **In the case of heterogeneous gestures**, specialized classifiers for different gesture types included in the dataset could be considered.
- **New augmentation methods should be proposed**, especially for the non-gesture movements, considering the biomechanical constraints.

6. Conclusion

In this study, we systematically surveyed the literature for continuous hand and upper body gesture recognition benchmarks and the methods providing the best performances on them. Our survey identified 12 continuous gesture recognition benchmarks published up to January 2024. For the best methods, we collected a total of 27 methods across the benchmarks. In our work, we dedicated special attention to the evaluation metrics proposed in the literature for the continuous recognition task, providing a taxonomy for them and discussing their critical aspects. It is evident, from our findings, that there are few benchmarks, with a number of them being recent and used in a low number of research works, often affected by relevant issues, as they are evaluated without using event-based metrics or they include datasets collected on a few subjects. Our analysis allows us to point out the specificity of the continuous recognition task and to compare the approaches recently proposed to deal with it.

Researchers who are new to the field can use this survey as a guide to becoming familiar with the terms, data types, evaluation metrics, methods, and benchmarks, but this work is not just limited to the new researchers. It also serves any researcher in academia and industry as it can suggest topics for new work and provide helpful guidance to interaction designers.

CRediT authorship contribution statement

Marco Emporio: Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualization. **Amirpouya Ghasemaghaei:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Joseph J. Laviola:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Conceptualization. **Andrea Giachetti:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Andrea Giachetti, Marco Emporio reports financial support was provided by European Union. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was partially carried out within the PNRR research activities of the consortium iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043).

Data availability

No data was used for the research described in the article.

References

Achanta, R., et al., 2009. Frequency-tuned salient region detection. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1597–1604. http://dx.doi.org/10.1109/CVPR.2009.5206596.

Amma, C., et al., 2015. Advancing muscle-computer interfaces with high-density electromyography. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 929–938.

Atzori, M., et al., 2014. Characterization of a benchmark database for myoelectric movement classification. IEEE Trans. Neural Syst. Rehabil. Eng. 23 (1), 73–83.

Benalcázar, M.E., et al., 2017. Hand gesture recognition using machine learning and the myo armband. In: 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, pp. 1040–1044.

Benitez-Garcia, G., et al., 2021a. Improving real-time hand gesture recognition with semantic segmentation. Sensors 21 (2), 356.

Benitez-Garcia, G., et al., 2021b. IPN hand: A video dataset and benchmark for real-time continuous hand gesture recognition. In: 2020 25th International Conference on Pattern Recognition. ICPR, IEEE, pp. 4340–4347.

Berg, J., Lu, S., 2020. Review of interfaces for industrial human-robot interaction. Curr. Robot. Rep. 1, 27–34.

Cabrera, M.E., Wachs, J.P., 2018. Biomechanical-based approach to data augmentation for one-shot gesture recognition. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, pp. 38–44.

Caeiro-Rodríguez, M., et al., 2021. A systematic review of commercial smart gloves: Current status and applications. Sensors 21 (8), 2667.

Caputo, F., et al., 2018. Comparing 3D trajectories for simple mid-air gesture recognition. Comput. Graph..

Caputo, F., et al., 2019. Shrec 2019 track: online gesture recognition. In: Eurographics Workshop on 3D Object Retrieval. The Eurographics Association, pp. 93–102.
Caputo, A., et al., 2021. SHREC 2021: Skeleton-based hand gesture recognition in the wild. Comput. Graph..
Chai, X., et al., 2016. Two streams recurrent neural networks for large-scale continuous gesture recognition. In: 2016 23rd International Conference on Pattern Recognition. ICPR, IEEE, pp. 31–36.
Chang, A., et al., 2014. GestureNet: a common sense approach to physical activity similarity. Electron. Vis. Arts (EVA 2014) 89–94.
Chao, P., et al., 2019. Hardnet: A low memory traffic network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3552–3561.
Cheng, H., et al., 2016. Survey on 3D hand gesture recognition. IEEE Trans. Circuits Syst. Video Technol. 26 (9), 1659–1673. http://dx.doi.org/10.1109/TCSVT.2015.2469551.
Cunico, F., et al., 2023. OO-dMVMT: A deep multi-view multi-task classification framework for real-time 3D hand gesture classification and segmentation. arXiv preprint arXiv:2304.05956.
De Smedt, Q., et al., 2016. Skeleton-based dynamic hand gesture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
De Smedt, Q., et al., 2017. Shrec'17 track: 3d hand gesture recognition using a depth and skeletal dataset. In: 3DOR-10th Eurographics Workshop on 3D Object Retrieval. pp. 1–6.
Dietterich, T.G., 2002. Machine learning for sequential data: A review. In: Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops SSPR 2002 and SPR 2002 Windsor, Ontario, Canada, August 6–9, 2002 Proceedings. Springer, pp. 15–30.
Doždor, Z., et al., 2023. Two-model-based online hand gesture recognition from skeleton data. In: Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 4: VISAPP. SciTePress, pp. 838–845. http://dx.doi.org/10.5220/0011663200003417, INSTICC.
Drossis, G., et al., 2013. MAGIC: developing a multimedia gallery supporting mid-air gesture-based interaction and control. In: HCI International 2013-Posters' Extended Abstracts: International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part I 15. Springer, pp. 303–307.
Emporio, M., et al., 2022. SHREC 2022 track on online detection of heterogeneous gestures. Comput. Graph..
Escalera, S., et al., 2015. ChaLearn looking at people challenge 2014: Dataset and results. In: Agapito, L., Bronstein, M.M., Rother, C. (Eds.), Computer Vision - ECCV 2014 Workshops. Springer International Publishing, Cham, pp. 459–473.
Escalera, S., et al., 2017. Challenges in multi-modal gesture recognition. Gesture Recognit. 1–60.
Gammulle, H., et al., 2021. TMMF: Temporal multi-modal fusion for single-stage continuous gesture recognition. IEEE Trans. Image Process. 30, 7689–7701. http://dx.doi.org/10.1109/TIP.2021.3108349.
Goswami, M., et al., 2018. A comparative analysis of similarity measures to find coherent documents. Appl. Sci. Manag. 8 (11), 786–797.
Gu, F., et al., 2021. A survey on deep learning for human activity recognition. ACM Comput. Surv. 54 (8), http://dx.doi.org/10.1145/3472290.
Guo, Q., Zhang, S., Li, H., 2023. Continuous sign language recognition based on spatial-temporal graph attention network. Comput. Model. Eng. Sci. 134, 1653–1670. http://dx.doi.org/10.32604/cmes.2022.021784.
Guyon, I., et al., 2012. Results and analysis of the ChaLearn gesture challenge 2012. In: Revised Selected and Invited Papers of the International Workshop on Advances in Depth Image Analysis and Applications. Vol. 7854, Springer-Verlag, Berlin, Heidelberg, pp. 186–204. http://dx.doi.org/10.1007/978-3-642-40303-3_19.
He, K., et al., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.
Huang, S., et al., 2020. To touch or not to touch? Comparing touch, mid-air gesture, mid-air haptics for public display in post COVID-19 society. In: SIGGRAPH Asia 2020 Posters. pp. 1–2.
Iandola, F.N., et al., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1 MB model size. arXiv preprint arXiv:1602.07360 arXiv:1602.07360.
Jain, R., et al., 2022. Literature review of vision-based dynamic gesture recognition using deep learning techniques. Concurr. Comput.: Pr. Exp. 34 (22), e7159.
Joze, H.R.V., et al., 2019. MMTM: Multimodal transfer module for CNN fusion. CoRR URL: http://arxiv.org/abs/1911.08670.
Kahol, K., et al., 2003. Gesture segmentation in complex motion sequences. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), vol. 2, IEEE, pp. II–105.
Kakkoth, S.S., Gharge, S., 2017. Survey on real time hand gesture recognition. In: 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication. CTCEEC, pp. 948–954. http://dx.doi.org/10.1109/CTCEEC.2017.8455041.
Karam, M., Schraefel, m., 2005. A taxonomy of gestures in human computer interactions. Electron. Comput. Sci..
Köpüklü, O., et al., 2020. Online dynamic hand gesture recognition including efficiency analysis. IEEE Trans. Biom. Behav. Identity Sci. 2 (2), 85–97. http://dx.doi.org/10.1109/TBIOM.2020.2968216.
LaViola, J.J., 2013. 3D gestural interaction: The state of the field. Int. Sch. Res. Not. 2013.
Lee, K.-S., Jung, M.-C., 2015. Ergonomic evaluation of biomechanical hand function. Saf. Heal. Work. 6 (1), 9–17.
Lee, S.-H., et al., 2022. Markerless 3D skeleton tracking algorithm by merging multiple inaccurate skeleton data from multiple RGB-D sensors. Sensors 22 (9), 3155.
Levenshtein, V.I., et al., 1966. Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, (8), Soviet Union, pp. 707–710.
Li, Y., et al., 2019a. Gesture interaction in virtual reality. Virtual Real. Intell. Hardw. 1 (1), 84–112. http://dx.doi.org/10.3724/SP.J.2096-5796.2018.0006.
Li, Y., et al., 2019b. Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. EURASIP J. Image Video Process. 2019 (1), 1–7.
Li, J., et al., 2019c. Weakly supervised energy-based learning for action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6243–6251.
Lin, J., et al., 2019. Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093.
Liu, D., Zhang, L., Wu, Y., 2022. LD-ConGR: A large RGB-D video dataset for long-distance continuous gesture recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3304–3312.
Maghoumi, M., LaViola, J.J., 2019. DeepGRU: Deep gesture recognition utility. In: Advances in Visual Computing: 14th International Symposium on Visual Computing, ISVC 2019, Lake Tahoe, NV, USA, October 7–9, 2019, Proceedings, Part I 14. Springer, pp. 16–31.
Maghoumi, M., et al., 2021. DeepNAG: Deep non-adversarial gesture generation. In: 26th International Conference on Intelligent User Interfaces. pp. 213–223.
Maslych, M., et al., 2023. Effective 2D Stroke-based gesture augmentation for RNNs. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–13.
Materzynska, J., et al., 2019. The jester dataset: A large-scale video dataset of human gestures. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.
Mitra, S., Acharya, T., 2007. Gesture recognition: A survey. IEEE Trans. Syst. Man, Cybern. Part C (Appl. Reviews) 37 (3), 311–324. http://dx.doi.org/10.1109/TSMCC.2007.893280.
Mo, G.B., et al., 2021. Gesture knitter: A hand gesture design tool for head-mounted mixed reality applications. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. CHI '21, Association for Computing Machinery, New York, NY, USA, http://dx.doi.org/10.1145/3411764.3445766.
Molchanov, P., et al., 2015. Multi-sensor system for driver's hand-gesture recognition. http://dx.doi.org/10.1109/FG.2015.7163132.
Molchanov, P., et al., 2016. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 4207–4215. http://dx.doi.org/10.1109/CVPR.2016.456.
Ohn-Bar, E., Trivedi, M.M., 2014. Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. IEEE Trans. Intell. Transp. Syst. 15 (6), 2368–2377. http://dx.doi.org/10.1109/TITS.2014.2337331.
Oudah, M., et al., 2020. Hand gesture recognition based on computer vision: A review of techniques. J. Imaging 6 (8), http://dx.doi.org/10.3390/jimaging6080073.
Page, M.J., et al., 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. Int. J. Surg. 88, 105906.
Papadopoulos, T., et al., 2021. Interactions in augmented and mixed reality: An overview. Appl. Sci. 11 (18), 8752.
Peral, M., et al., 2022. Efficient hand gesture recognition for human-robot interaction. IEEE Robot. Autom. Lett. 7 (4), 10272–10279.
Pigou, L., et al., 2018. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. International Journal of Computer Vision 126, 430–439, URL: http://arxiv.org/abs/1506.01911.
Pisharady, P.K., Saerbeck, M., 2015. Recent methods and databases in vision-based hand gesture recognition: A review. Comput. Vis. Image Underst. 141, 152–165.
Prabhakar, G., Biswas, P., 2021. A brief survey on interactive automotive UI. Transp. Eng. 6, 100089.
Qiu, Z., Yao, T., Mei, T., 2017. Learning spatio-temporal representation with pseudo-3D residual networks. In: 2017 IEEE International Conference on Computer Vision. ICCV, pp. 5534–5542. http://dx.doi.org/10.1109/ICCV.2017.590.
Rautaray, S.S., Agrawal, A., 2015. Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. 43, 1–54.
Rocha, G., Cardoso, H.L., 2004. Exploring spanish corpora for portuguese coreference resolution. ACE 2000, 22.
Ruffieux, S., et al., 2013. ChAirGest: A challenge for multimodal mid-air gesture recognition for close HCI. In: Proceedings of the 15th ACM on International Conference on Multimodal Interaction. ICMI '13, Association for Computing Machinery, New York, NY, USA, pp. 483–488. http://dx.doi.org/10.1145/2522848.2532590.
Sarma, D., Bhuyan, M.K., 2021. Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review. SN Comput. Sci. 2 (6), 436.
Schäfer, A., et al., 2022. Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications. Appl. Sci. 12 (4), 1888.

Schneider, D., et al., 2021. Accuracy evaluation of touch tasks in commodity virtual and augmented reality head-mounted displays. In: Proceedings of the 2021 ACM Symposium on Spatial User Interaction. pp. 1–11.

Shen, J., et al., 2021. The imaginative generative adversarial network: Automatic data augmentation for dynamic skeleton-based hand gesture and human action recognition. In: 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021). IEEE, pp. 1–8.

Shen, J., et al., 2022. Gesture spotter: A rapid prototyping tool for key gesture spotting in virtual and augmented reality applications. IEEE Trans. Vis. Comput. Graphics 28 (11), 3618–3628.

Shi, L., et al., 2020. Decoupled spatial-temporal attention network for skeleton-based action-gesture recognition. In: Proceedings of the Asian Conference on Computer Vision.

Shi, Y., et al., 2021. Review of dynamic gesture recognition. Virtual Real. Intell. Hardw. 3 (3), 183–206. http://dx.doi.org/10.1016/j.vrih.2021.05.001.

Tang, D., et al., 2014. Latent regression forest: Structured estimation of 3d articulated hand posture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3786–3793.

Tran, D., et al., 2014. C3D: generic features for video analysis. CoRR URL: http://arxiv.org/abs/1412.0767.

Tran, D., et al., 2015. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4489–4497.

Tran, D., et al., 2017. ConvNet architecture search for spatiotemporal feature learning. CoRR URL: http://arxiv.org/abs/1708.05038.

Tsironi, E., et al., 2017. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. Neurocomputing 268, 76–86.

Ungureanu, D., et al., 2020. Hololens 2 research mode as a tool for computer vision research. arXiv preprint arXiv:2008.11239.

Vandersteegen, M., et al., 2020. Low-latency hand gesture recognition with a low resolution thermal imager. CoRR URL: https://arxiv.org/abs/2004.11623.

Vatavu, R.-D., 2012. User-defined gestures for free-hand TV control. In: Proceedings of the 10th European Conference on Interactive Tv and Video. pp. 45–48.

Wan, J., et al., 2016. ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops. CVPRW, pp. 761–769. http://dx.doi.org/10.1109/CVPRW.2016.100.

Wang, H., 2021. Two stage continuous gesture recognition based on deep learning. Electronics 10 (5), http://dx.doi.org/10.3390/electronics10050534.

Wang, L., et al., 2016. Temporal segment networks: Towards good practices for deep action recognition. In: European Conference on Computer Vision. Springer, pp. 20–36.

Wannous, H., Vandeborre, J.-P., 2022. Continuous hand gesture recognition using deep coarse and fine hand features. In: The 33rd British Machine Vision Conference–BMVC 2022.

Ward, J.A., et al., 2011. Performance metrics for activity recognition. ACM Trans. Intell. Syst. Technol. 2 (1), http://dx.doi.org/10.1145/1889681.1889687.

Wu, P., et al., 2020. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11385–11395.

Xie, S., et al., 2017. Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 5987–5995. http://dx.doi.org/10.1109/CVPR.2017.634.

Xu, C., et al., 2023. Improving dynamic gesture recognition in untrimmed videos by an online lightweight framework and a new gesture dataset ZJUGesture. Neurocomputing 523, 58–68.

Yang, F., et al., 2019. Make skeleton-based action recognition model smaller, faster and better. In: Proceedings of the ACM Multimedia Asia. pp. 1–6.

Yin, Y., Davis, R., 2013. Gesture spotting and recognition using salience detection and concatenated hidden Markov models. In: Proceedings of the 15th ACM on International Conference on Multimodal Interaction. ICMI '13, Association for Computing Machinery, New York, NY, USA, pp. 489–494. http://dx.doi.org/10.1145/2522848.2532588.

Zhang, Y., et al., 2018. EgoGesture: A new dataset and benchmark for egocentric hand gesture recognition. IEEE Trans. Multimed. 20 (5), 1038–1050. http://dx.doi.org/10.1109/TMM.2018.2808769.

Zhang, F., et al., 2020. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.

Zhu, G., et al., 2017. Multimodal gesture recognition using 3-D convolution and convolutional LSTM. Ieee Access 5, 4517–4524.

Zhu, G., et al., 2019. Continuous gesture segmentation and recognition using 3DCNN and convolutional LSTM. IEEE Trans. Multimed. 21 (4), 1011–1021. http://dx.doi.org/10.1109/TMM.2018.2869278.