Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.045731

REVIEW

Recent Advances on Deep Learning for Sign Language Recognition

Yanqiong Zhang and Xianwei Jiang*

School of Mathematics and Information Science, Nanjing Normal University of Special Education, Nanjing, 210038, China

*Corresponding Author: Xianwei Jiang. Email: jxw@njts.edu.cn

Received: 06 September 2023 Accepted: 21 November 2023 Published: 11 March 2024

ABSTRACT

Sign language, a visual-gestural language used by the deaf and hard-of-hearing community, plays a crucial role in facilitating communication and promoting inclusivity. Sign language recognition (SLR), the process of automatically recognizing and interpreting sign language gestures, has gained significant attention in recent years due to its potential to bridge the communication gap between the hearing impaired and the hearing world. The emergence and continuous development of deep learning techniques have provided inspiration and momentum for advancing SLR. This paper presents a comprehensive and up-to-date analysis of the advancements, challenges, and opportunities in deep learning-based sign language recognition, focusing on the past five years of research. We explore various aspects of SLR, including sign data acquisition technologies, sign language datasets, evaluation methods, and different types of neural networks. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown promising results in fingerspelling and isolated sign recognition. However, the continuous nature of sign language poses challenges, leading to the exploration of advanced neural network models such as the Transformer model for continuous sign language recognition (CSLR). Despite significant advancements, several challenges remain in the field of SLR. These challenges include expanding sign language datasets, achieving user independence in recognition systems, exploring different input modalities, effectively fusing features, modeling co-articulation, and improving semantic and syntactic understanding. Additionally, developing lightweight network architectures for mobile applications is crucial for practical implementation. By addressing these challenges, we can further advance the field of deep learning for sign language recognition and improve communication for the hearing-impaired community.

KEYWORDS

Sign language recognition; deep learning; artificial intelligence; computer vision; gesture recognition

1 Introduction

Effective communication is essential for individuals to express their thoughts, feelings, and needs. However, for individuals with hearing impairments, spoken language may not be accessible. In such cases, sign language serves as a vital mode of communication. Sign language is a visual-gestural language that utilizes hand movements, facial expressions, and body postures to convey meaning. This unique language has a rich history and has evolved to become a distinct and complex system of

communication. Sign languages differ across regions and countries, with each having its own grammar and vocabulary.

Stokoe W. C. made a significant contribution to the understanding of sign language by recognizing its structural similarities to spoken languages. Like spoken languages, sign language has a phonological system. Signs can be broken down into smaller linguistic units [1]. As shown in Fig. 1, sign language can be categorized into manual and non-manual features. Manual features can be further divided into handshape, orientation, position, and movement. Non-manual features include head and body postures, and facial expressions. These features work together to convey meaning and enable effective communication in sign language.

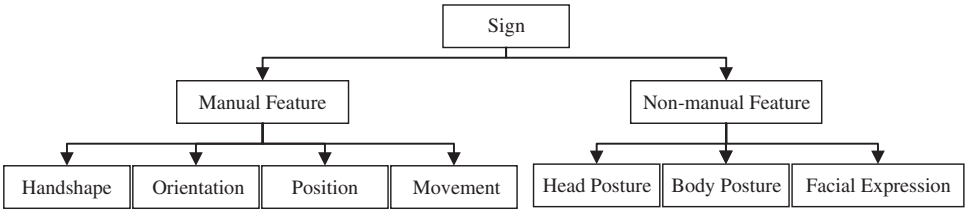

Figure 1: Structural features of sign language

According to the World Health Organization, there are over 466 million people globally with disabling hearing loss, and this number is expected to increase in the coming years. For individuals who are deaf or hard of hearing, sign language is often their primary mode of communication. However, the majority of the population does not understand sign language, leading to significant communication barriers and exclusion for the deaf community. Sign language recognition (SLR) refers to the process of automatically interpreting and understanding sign language gestures and movements through various technological means, such as computer vision and machine learning algorithms. By enabling machines to understand and interpret sign language, we can bridge the communication gap between the deaf community and the hearing world. SLR technology has the potential to revolutionize various sectors, including education, healthcare, and communication, by empowering deaf individuals to effectively communicate and access information, services, and opportunities that were previously limited [2,3]. In addition, SLR technology can be expanded to other areas related to gesture commands, such as traffic sign recognition, military gesture recognition, and smart appliance control [4–8].

Research on SLR dates back to the 1990s. Based on the nature of the signs, these techniques were categorized into fingerspelling recognition, isolated sign language recognition, and continuous sign language recognition, as depicted in Fig. 2.

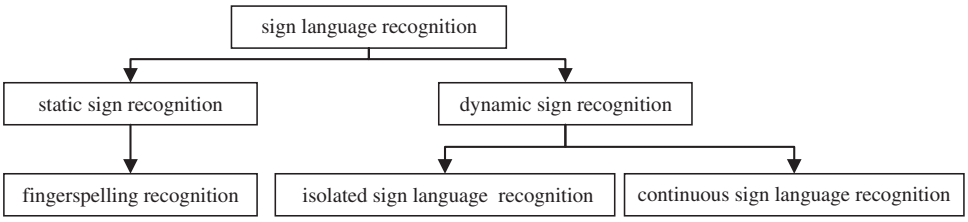

Figure 2: Sign language recognition classification

Static signs, such as alphabet and digit signs, primarily belong to the category of fingerspelling recognition. This type of recognition involves analyzing and interpreting the specific hand shapes and

positions associated with each sign. Although it is important to acknowledge that certain static signs may involve slight movements or variations in hand shape, they are generally regarded as static because their main emphasis lies in the configuration and positioning of the hands rather than continuous motion.

On the other hand, dynamic signs can be further classified into isolated sign recognition and continuous sign recognition systems. Isolated sign gesture recognition aims to recognize individual signs or gestures performed in isolation. It involves identifying and classifying the hand movements, facial expressions, and other relevant cues associated with each sign. In contrast, continuous sign recognition systems aim to recognize complete sentences or phrases in sign language. They go beyond recognizing individual signs and focus on understanding the context, grammar, and temporal sequence of the signs. This type of recognition is crucial for facilitating natural and fluid communication in sign language.

In the field of sign language recognition, traditional machine learning methods have played significant roles. These methods have been utilized for feature extraction, classification, and modeling of sign language. However, traditional machine learning approaches often face certain limitations and have reached a bottleneck. These limitations include the need for manual feature engineering, which can be time-consuming and may not capture all the relevant information in the data. Additionally, these methods may struggle with handling complex and high-dimensional data, such as the spatiotemporal information present in sign language gestures. Over recent years, deep learning methods outperformed previous state-of-the-art machine learning techniques in different areas, especially in computer vision and natural language processing [9]. Deep learning techniques have brought significant advancements to sign language recognition [10–14], leading to a surge in research papers published on deep learning-based SLR. As the field continues to evolve, it is crucial to conduct updated literature surveys. Therefore, this paper aims to provide a comprehensive review and classification of the current state of research in deep learning-based SLR.

This review delves into various aspects and technologies related to SLR using deep learning, covering the latest advancements in the field. It also discusses publicly available datasets commonly used in related research. Additionally, the paper addresses the challenges encountered in SLR and identifies potential research directions. The remaining sections of the paper are organized as follows: Section 2 describes the collection and quantitative analysis of literature related to SLR. Section 3 describes different techniques for acquiring sign language data. Section 4 discusses sign language datasets and evaluation methods. Section 5 explores deep learning techniques relevant to SLR. In Section 6, advancements and challenges of various techniques employed in SLR are compared and discussed. Finally, Section 7 summarizes the development directions in this field.

2 Literature Collection and Quantitative Analysis

In this study, we conducted a systematic review following the Preferred Reporting Items for Systematic Review and Meta-Analysis (PRISMA). We primarily rely on the Web of Science Core Collection as our main source of literature. This database encompasses a wide range of high-quality journals and international conferences. Additionally, we have also supplemented our research by searching for relevant literature on sign language datasets and the application of SLR in embedded systems in other databases. To ensure the relevance of our research, we applied specific selection criteria, focusing on peer-reviewed articles and proceeding papers published between 2018 and 2023 (Search date: 2023-06-12). Our review targeted sign language recognition in deep learning and also included papers related to sign language translation, considering its two-step process involving continuous

sign language recognition (CSLR) and gloss-to-text translation. After eliminating irrelevant papers, our study encompassed 346 relevant papers. The PRISMA chart depicting our selection process is presented in Fig. 3.

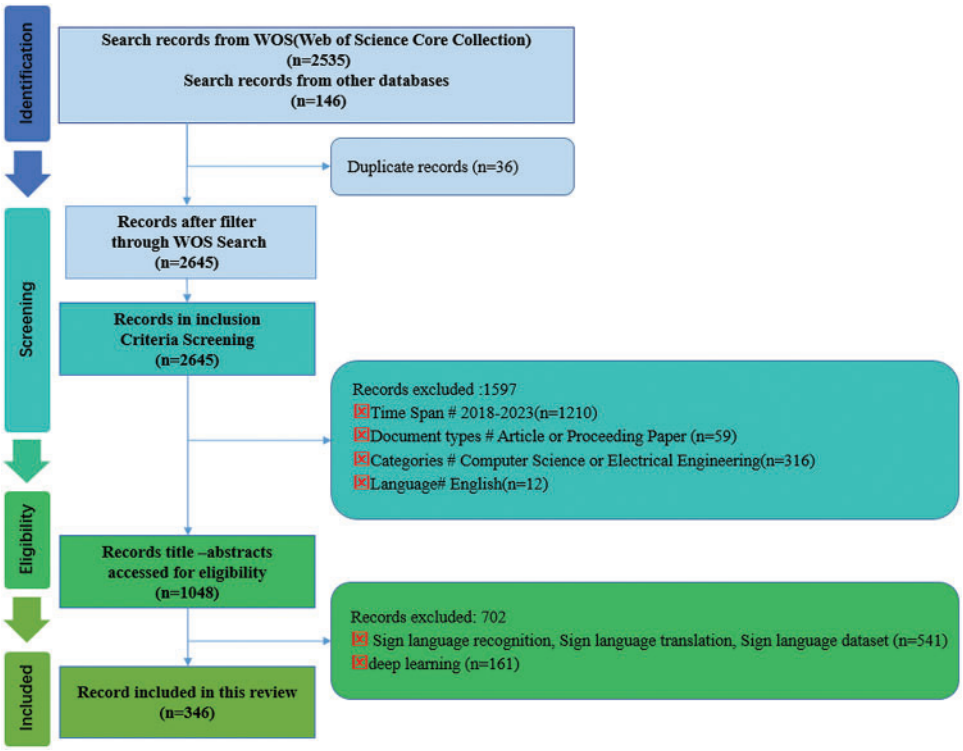

Figure 3: The PRISMA flow diagram for identifying relevant documents included in this review

A comprehensive literature analysis was performed on various aspects of SLR using deep learning, including annual publication volume, publishers, sign language subjects, main technologies, and architectures. Fig. 4 demonstrates a consistent increase in the number of publications each year, indicating the growing interest and continuous development in this field. Fig. 5 highlights the prominent publishers in the domain of deep learning-based SLR. Notably, IEEE leads with the highest number of publications, accounting for 37.57% of the total, followed by Springer Nature with 19.36% and Mdpi with 10.41%. Table 1 displays the primary sign language subjects for research, encompassing American SL, Indian SL, Chinese SL, German SL, and Arabic SL. It is important to note that this data is derived from the experimental databases utilized in the papers. In cases where a paper conducted experiments using multiple databases, each database is counted individually. For instance, experiments were conducted on two test datasets: the RWTH-PHOENIX-Weather multi-signer dataset and a Chinese SL (CSL) dataset [15]. Therefore, German SL and Chinese SL are each counted once. Table 2 presents the main technologies and architectures employed in deep learning-based SLR. In Session 5, we will focus on elucidating key technological principles to facilitate comprehension for readers new to this field. The statistical data in Table 2 is obtained by first preprocessing and normalizing the keywords in the literature, and then using VOSviewer software to analyze and calculate the keywords.

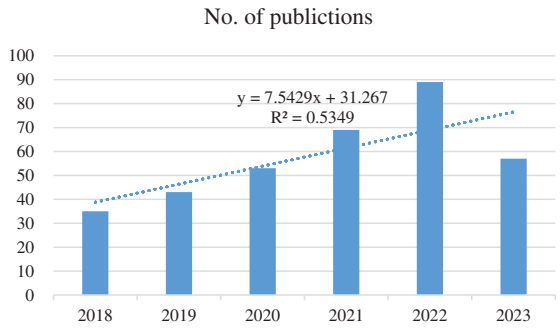

Figure 4: Number of publications on sign language recognition by year

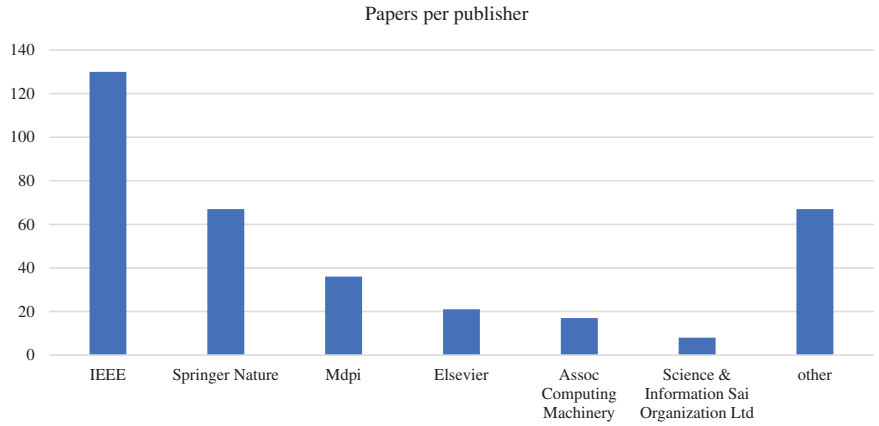

Figure 5: Number of publications on sign language recognition by publisher

Table 1: The main sign language subjects on sign language recognition in deep learning (No. >=5)

Sign language	No.	Sign language	No.
American SL	54	Turkish SL	6
Indian SL	35	British SL	6
German SL	33	Japanese SL	6
Chinese SL	26	Pakistan SL	5
Arabic SL	25	Russian SL	5
Korean SL	11	Bangla SL	5

Table 2: The main technologies or architectures of sign language recognition in deep learning (No. >=5)

Technologies (Architectures)	No.	Technologies (Architectures)	No.
CNN	91	ResNet	8
Transfer learning	33	vgg	8

(Continued)

Table 2 (continued)

Technologies (Architectures)	No.	Technologies (Architectures)	No.
Attention	27	CNN-LSTM	8
Transformer	20	CNN-HMM	7
LSTM	16	yolo	7
3D-CNN	15	Ensemble learning	6
Inception	11	Generative adversarial networks	5
RNN	11	Lightweight network	5
Bi-LSTM	8	Mobilenet	5
Graph convolutional network	8	gru	5

3 Sign Data Acquisition Technologies

The techniques used to acquire sign language can be categorized into sensor-based, vision-based, or a combination of both. Sensor-based techniques rely on a variety of sensors to capture and record the gesture information of signers, while vision-based techniques utilize video cameras to capture the gestures performed by signers.

3.1 Sensor-Based Acquisition Technologies

With the development of electronic science and technology, sensors have garnered significant attention and found applications in various fields. In the context of sign language, sensors play a crucial role in measuring and recording data related to hand movements, including bending, shape, rotation, and position during the signing process. Several approaches exist for acquiring sign language data using sensors, such as strain sensors, surface electromyography (sEMG) sensors, tactile or pressure sensors, as well as inertial sensors like accelerometers, magnetometers, and gyroscopes. Technological progress has led to the development of compact and affordable sensors, microcontrollers, circuit boards, and batteries. These portable systems have the capacity to store large amounts of sensor data. However, a drawback of sensor-based approaches is the potential discomfort or restriction of movement caused by the sensor configuration. To address this issue, sensors can be integrated into wearable devices such as digital gloves, wristbands, or rings [16].

Digital gloves are one of the most common sensor-based devices for acquiring sign language data. Fig. 6 illustrates examples of these glove sensors. The sensors attached to digital gloves for sign language data acquisition can be categorized into those measuring finger bending and those measuring hand movement and orientation [17]. Lu et al. [18] developed the Sign-Glove, which consisted of a bending sensor and an inertial measurement unit (IMU). The IMU captured the motion features of the hand, as shown in Fig. 7b, while the bending sensor collected the bending (shape) features of the hand, as shown in Fig. 7c. Alzubaidi et al. [19] proposed a novel assistive glove that converts Arabic sign language into speech. This glove utilizes an MPU6050 accelerometer/gyro with 6 degrees of freedom to continuously monitor hand orientation and movement. DelPreto et al. [20] presented the smart multi-modal glove system, which integrated an accelerometer to capture both hand pose and gesture information based on a commercially available glove. Oz et al. [21] used Cyberglove and a Flock of Bird 3-D motion tracker to extract the gesture features. The Cyberglove is equipped with 18 sensors to measure the bending angles of fingers, while the Flock of Bird 3-D motion tracker tracks the position and orientation of the hand. Dias et al. [22] proposed an instrumented glove

with five flex sensors, an inertial sensor, and two contact sensors for recognizing the Brazilian sign language alphabet. Wen et al. [23] utilized gloves configured with 15 triboelectric sensors to track and record hand motions such as finger bending, wrist motions, touch with fingertips, and interaction with the palm.

(a) a wearable smart glove (b) CybergloveTM with 18 sensors (c) an instrumented glove

Figure 6: Examples of these glove sensors

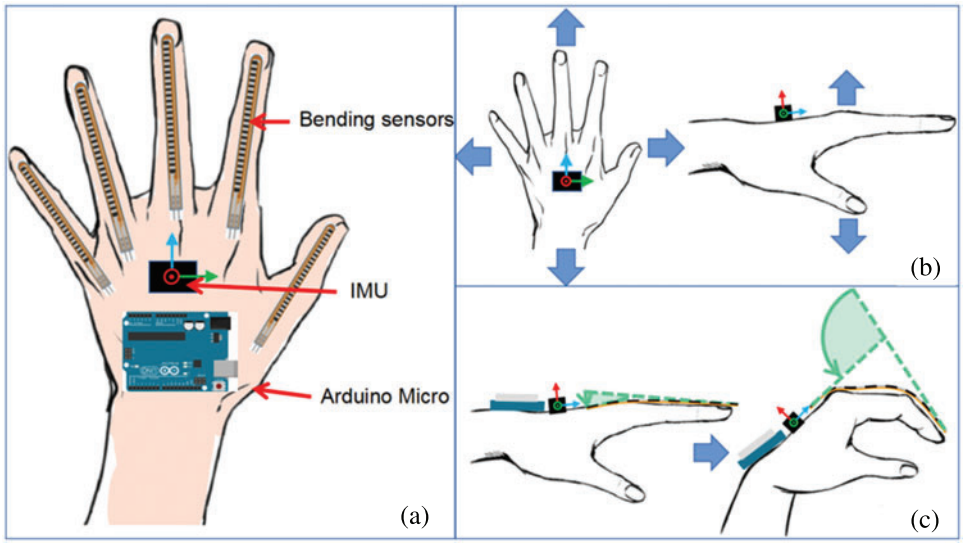

Figure 7: (a) System structure. (b) IMU collecting the hand motion data. (c) Bending sensor collecting the hand shape data

Leap Motion Controller (LMC) is a small, motion-sensing device that allows users to interact with their computer using hand and finger gestures. It uses infrared sensors and cameras to track the movement of hands and fingers in 3D space with high precision and accuracy. In the field of sign

language recognition, by tracking the position, orientation, and movement of hands and fingers, the Leap Motion Controller can provide real-time data that can be used to recognize and interpret sign language gestures [24–26].

Some studies have utilized commercially available devices such as the Myo armband [27–29], which are worn below the elbow and equipped with sEMG and inertial sensors. The sEMG sensors can measure the electrical potentials produced by muscles. By placing these sensors on the forearm over key muscle groups, specific hand and finger movements can be identified and recognized [30–32]. Li et al. [27] used A wearable Myo armband to collect human arm surface electromyography (sEMG) signals for improving SLR accuracy. Pacifici et al. [28] built a comprehensive dataset that includes EMG and IMU data captured with the Myo Gesture Control Armband. This data was collected while performing the complete set of 26 gestures representing the alphabet of the Italian Sign Language. Mendes Junior et al. [29] demonstrated the classification of a series of alphabet gestures in Brazilian Sign Language (Libras) through the utilization of sEMG obtained from a Myo™ armband.

Recent literature studies have highlighted the potential of utilizing WiFi sensors to accurately identify hand and finger gestures through channel state information [33–36]. The advantage of WiFi signals is their nonintrusive nature, allowing for detachment from the user's hand or finger and enabling seamless recognition. Zhang et al. [33] introduced a WiFi-based SLR system called Wi-Phrase, which applies principal component analysis (PCA) projection to eliminate noise and transform cleaned WiFi signals into a spectrogram. Zhang et al. [35] proposed WiSign, which recognizes continuous sentences of American Sign Language (ASL) using existing WiFi infrastructure. Additionally, RF sensors provide a pathway for SLR [37–41].

Sensor-based devices offer the benefit of minimizing reliance on computer vision techniques for signer body detection and segmentation. This allows the recognition system to identify sign gestures with minimal processing power. Moreover, these devices can track the signer's movements, providing valuable spatial and temporal information about the executed signs. However, it is worth noting that certain devices, such as digital gloves, require the signer to wear the sensor device while signing, limiting their applicability in real-time scenarios.

3.2 Vision-Based Acquisition Technologies

In vision-based sign acquisition, video cameras and other imaging devices are used to capture sign gestures and store them as sequences of images for further processing. Typically, three types of outputs are generated: color images, depth images, and skeleton images. Color images are obtained by using standard RGB cameras to capture the visible light spectrum. They provide detailed information about the appearance and color of the signer's gestures. However, they can be sensitive to lighting conditions and prone to noise from shadows or reflections, which may affect accuracy. Depth images are generated through depth-sensing technologies like Time-of-Flight (ToF) or structured light. These images provide 3D spatial information, offering a better understanding of the signer's 3D structure and spatial relationships. Skeleton images, on the other hand, are produced by extracting joint positions and movements from depth images or similar depth-based techniques. Like depth images, skeleton images also provide 3D spatial information, enabling a deeper understanding of the signer's body movements and posture. The combination of color, depth, and skeleton images in vision-based acquisition allows for a comprehensive analysis of sign gestures, providing valuable information about appearance, spatial structure, and movements.

Microsoft Kinect is a 3D motion-sensing device commonly used in gaming. It consists of an RGB camera, an infrared emitter, and an infrared depth sensor, which enables it to track the movements

of users. In the field of sign language recognition, the Kinect has been widely utilized [42–45]. Raghuveera et al. [46] captured hand gestures through Microsoft Kinect. Gangrade et al. [47,48] have leveraged the 3D depth information obtained from hand motions, which is generated by Microsoft's Kinect sensor.

Multi-camera and 3D systems can mitigate certain environmental limitations but introduce a higher computational burden, which can be effectively addressed due to the rapid progress in computing technologies. Kraljević et al. [49] proposed a high-performance sign recognition module that utilizes the 3DCNN network. They employ the StereoLabs ZED M stereo camera to capture real-time RGB and depth information of signs.

In SLR systems, vision-based techniques are more suitable compared to sensor-based approaches. These techniques utilize video cameras instead of sensors, eliminating the need for attaching sensors to the signer's body and overcoming the limited operating range of sensor-based devices. Vision-based devices, however, provide raw video streams that often require preprocessing for convenient feature extraction, such as signer detection, background removal, and motion tracking. Furthermore, computer vision systems must address the significant variability and sources of errors inherent in their operation. These challenges include noise and environmental factors resulting from variations in illumination, viewpoint, orientation, scale, and occlusion.

4 Sign Language Datasets and Evaluation Methods

Sign Language Datasets and Evaluation Methods are essential components in the development and advancement of SLR systems. They provide researchers with the necessary resources and tools to train, evaluate, and improve deep learning models for accurate and reliable SLR.

4.1 Sign Language Datasets

A sign language dataset is crucial for training and evaluating SLR systems. Factors for a good dataset include diversity of gestures, sufficient sample size, variations in conditions, inclusion of different users, and accurate annotations. Table 3 provides an overview of the most commonly used sign language databases, categorized by sign level (fingerspelling, isolated, and continuous). These databases are specifically designed for research purposes.

Table 3: The most common databases used for SLR

Dataset	Year	Language	Type	Signs/ vocabulary	Signers	Samples/ videos	Modality	Data_link
ASL Fingerspelling A [50]	2011	ASL	Fingerspelling	24	5	131000	RGB, D (Kinect)	[51]
PHOENIX 14 Handshapes	2014	GSL	Fingerspelling	60	–	3359	RGB	[52]
ASL Fingerspelling	2018	ASL	Fingerspelling	24	9	–	Myo sensor	[53]
ChicagoFSWild [54]	2018	ASL	Fingerspelling	26	160	7304	RGB	[55]
ChicagoFSWild+ [54]	2018	ASL	Fingerspelling	26	260	55,232	RGB	[55]
CSL Fingerspelling [56]	2019	CSL	Fingerspelling	30	40	1320	RGB	–
ArSL2018 [57]	2018	ArSL	Fingerspelling	32	40	54049	RGB	[58]
ArSL21L [2]	2021	ArSL	Fingerspelling	32	50	14202		–
MS-ASL-100 [59]	2018	ASL	Signs	100	189	5736	RGB	[60]
MS-ASL-200 [59]	2018	ASL	Signs	200	196	9719	RGB	[60]

(Continued)

Table 3 (continued)

Dataset	Year	Language	Type	Signs/ vocabulary	Signers	Samples/ videos	Modality	Data_link
MS-ASL-500 [59]	2018	ASL	Signs	500	222	17823	RGB	[60]
MS-ASL-1000 [59]	2018	ASL	Signs	1000	222	25513	RGB	[60]
CSL-500 [61]	2019	CSL	Signs	500	50	125000	RGB, D, skeleton	[62]
INCLUDE [63]	2020	ISL	Signs	263	–	4287	RGB	[64]
WLASL100 [65]	2020	ASL	Signs	100	97	2038	RGB	[66]
WLASL300 [65]	2020	ASL	Signs	300	109	5117	RGB	[66]
WLASL1000 [65]	2020	ASL	Signs	1000	116	13168	RGB	[66]
WLASL2000 [65]	2020	ASL	Signs	2000	119	21083	RGB	[66]
AUTSL [67]	2020	TuSL	Signs	226	42	38336	RGB	[68]
Libras [69]	2021	BrSL	Signs	20	–	1200	RGB, D, body points, face information	[70]
KArSL [71]	2021	ArSL	Signs	502	3	–	RGB, D, skeleton	[72]
BdSLW-11 [73]	2022	BdSL	Signs	11	–	1105	RGB	[74]
PHOENIX [75]	2012	GSL	Sentences	911	7	1980	RGB	[52]
PHOENIX 14 [76]	2014	GSL	Sentences	1080	9	6841	RGB	[52]
PHOENIX 14T [77]	2018	GSL	Sentences	1066	9	8257	RGB	[52]
CSL [78]	2018	CSL	Sentences	178	50	25000	RGB, D, body joints	[79]
SIGNUM [80]	2009	GSL	Sentences	455	25	33210	RGB	[81]
HKSL [11]	2022	HKSL	Sentences	50	6	–	RGB,D, smart watch data	–

Fingerspelling datasets primarily focus on sign language alphabets and/or digits. Some exclude letters that involve motion, such as ‘j’ and ‘z’ in American Sign Language [50]. The impact of signer variability on recognition systems is minimal in fingerspelling databases since most images only display the signer’s hands. These datasets mainly consist of static images as the captured signs do not involve motion [61,64,65].

Isolated sign datasets are the most widely used type of sign language datasets. They encompass isolated sign words performed by one or more signers. Unlike fingerspelling databases, these databases contain motion-based signs that require more training for non-expert signers. Vision-based techniques, such as video cameras, are commonly used to capture these signs [66,72,82]. However, there are other devices, like the Kinect, which output multiple data streams for collecting sign words [71,69].

Continuous sign language databases comprise a collection of sign language sentences, where each sentence consists of a continuous sequence of signs. This type of database presents more challenges compared to the previous types, resulting in a relatively limited number of available databases. Currently, only PHOENIX14 [76], PHOENIX14T [77] and CSL Database [78] are used regularly. The scarcity of sign language datasets suitable for CSLR can be attributed to the time-consuming and complex nature of dataset collection, the diversity of sign languages, and the difficulty of annotating the data.

Upon analysis, it is evident that current sign language datasets have certain limitations.

(1) Small scale and limited vocabulary

These datasets are often small in scale and have a limited vocabulary. This poses a challenge for training models to recognize a wide range of signs and gestures accurately. A larger and more diverse dataset with a comprehensive vocabulary would enable better performance and generalization of SLR systems.

(2) Lack of diversity in terms of the signers included

Sign language users come from various backgrounds, cultures, and regions, leading to variations in signing styles, handshapes, and facial expressions. By including a more diverse set of signers, the datasets can better represent the richness and complexity of sign languages, improving the robustness and accuracy of recognition systems.

(3) Lack of variability in controlled environments

Most existing datasets are collected in controlled settings, such as laboratories or studios, which may not accurately reflect real-world scenarios. Sign language is often used in various contexts, including different lighting conditions, backgrounds, and levels of noise. Incorporating such variability in dataset collection would enhance the robustness of SLR models, enabling them to perform well in real-world applications.

(4) Lack of multimodal data

The current sign language datasets often lack multimodal data, which includes video, depth information, and skeletal data. By incorporating multimodal data, the datasets can capture the full range of visual cues used in sign language and improve the accuracy and naturalness of recognition systems.

4.2 Evaluation Method

Evaluating the performance of SLR techniques is crucial to assess their effectiveness. Several evaluation methods are commonly used in this domain.

4.2.1 Evaluation Method for Isolated SLR and Fingerspelling Recognition

Methods for evaluating Isolated SLR and fingerspelling recognition generally include Accuracy, Precision, Recall, and F1-Score.

Accuracy measures the overall correctness of the model's predictions and is calculated as the ratio of the number of correctly classified instances to the total number of instances. Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It focuses on the model's ability to avoid false positives. Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances out of all actual positive instances. It focuses on the model's ability to avoid false negatives. Their calculation formulas are as follows:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

$$Recall = TP/(TP + FN) \tag{3}$$

$$F1 - Score = 2 * (Precision * Recall/(Precision + Recall)) \tag{4}$$

To understand these metrics, let us break down the components:

- True Positive (*TP*) represents the number of correctly recognized positive gestures.
- True Negative (*TN*) represents the number of correctly recognized negative gestures.
- False Positive (*FP*) represents the number of negative gestures incorrectly classified as positive.
- False Negative (*FN*) represents the number of positive gestures incorrectly classified as negative.

4.2.2 Evaluation Method for CSLR

Methods for evaluating CSLR generally include Word Error Rate (WER) and Accuracy. However, it is important to note that Accuracy in CSLR is not calculated using the same formula as mentioned previously. WER calculates the minimum number of insertions, deletions, and substitutions needed to transform the recognized sequence into the ground truth sequence, divided by the total number of words or gestures in the ground truth sequence. The formula to calculate WER is as follows:

$$WER = \frac{n_s + n_d + n_i}{L} \times 100\% \tag{5}$$

Here, n_s represents the number of substitutions. n_d represents the number of deletions. n_i represents the number of insertions. L represents the total number of words or gestures in the ground truth sequence.

Accuracy and WER are a pair of opposite concepts, and their calculation method is as follows:

$$\text{Accuracy} = 1 - WER \tag{6}$$

In CSLR, a higher WER indicates a lower level of Accuracy and a lower WER indicates a higher level of Accuracy.

5 Deep Learning–Based SLR Approach

The development of deep learning techniques has greatly advanced the field of SLR. In the following section, we will introduce some commonly used deep learning techniques in the field of sign language recognition.

5.1 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that excels in processing visual data, making them highly suitable for analyzing sign language gestures. In the field of sign language recognition, CNNs have gained widespread use.

The main components of CNNs include convolutional layers, pooling layers, and fully connected layers. The simplified structure diagram of CNNs is depicted in Fig. 8.

Input Layer: This layer receives the raw image as input.

Convolutional Layer: By applying a series of filters (convolutional kernels) to the input image, this layer performs convolution operations to extract local features. Each filter generates a feature map.

2D convolution is the fundamental operation in CNNs. It involves sliding a 2D filter (also known as a kernel) over the input image, performing element-wise multiplication between the filter and the local regions of the image, and summing the results. This operation helps in capturing local spatial dependencies and extracting spatial features. The following takes an image as an example to show 2D convolution, as shown in Fig. 9.

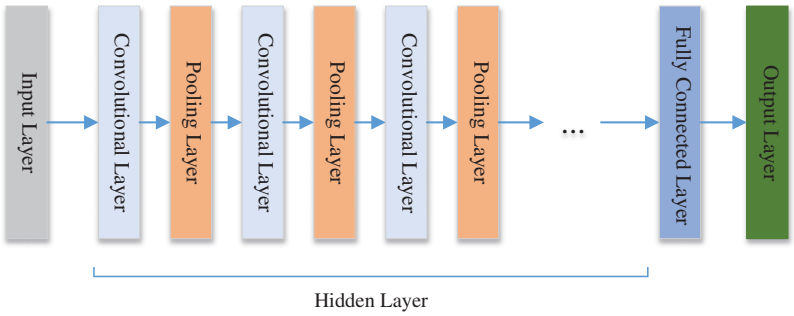

Figure 8: A simple CNN diagram

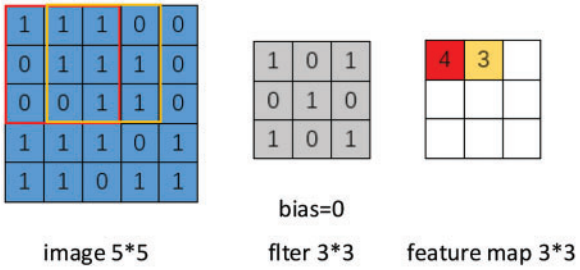

Figure 9: Illustration of 2D convolution

The calculation process of the first value of the feature map is as follows:

$$feature\ map\ (1) = 1 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 0 + 1 \times 1 + 0 \times 1 + 1 \times 0 + 0 \times 0 + 1 \times 1 = 4$$

$$feature\ map\ (2) = 1 \times 1 + 0 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 1 + 0 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 1 = 3$$

Fig. 9 illustrates the convolution operation with an input. Assuming the input image has a shape of $H_{in} \times W_{in}$, convolution kernels is with a shape of $K_h \times K_w$. Convolution is performed using a kernel of size $K_h \times K_w$ on a 2D array of size $H_{in} \times W_{in}$. The results are then summed, resulting in a 2D array with a shape of $H_{out} \times W_{out}$. The formula to calculate the size of the output feature map is as follows:

$$H_{out} = \frac{H_{in} - K_h + 2 \times P_h}{S_h} + 1 \tag{7}$$

$$W_{out} = \frac{W_{in} - K_w + 2 \times P_w}{S_w} + 1 \tag{8}$$

Here, S_h and S_w represent the stride value in the vertical and horizontal directions, respectively. P_h and P_w denote the size of padding in the vertical and horizontal directions, respectively.

Unlike traditional 2D CNNs, which are primarily employed for image analysis, 3D CNNs are designed specifically for the analysis of volumetric data. This could include video sequences or medical scans. 3D CNNs model utilizes 3D convolutions to extract features from both spatial and temporal dimensions. This allows the model to capture motion information encoded within multiple adjacent frames [83]. This allows for enhanced feature representation and more accurate decision-making in tasks such as action recognition [10,84,85], video segmentation [86], and medical image analysis [87].

Activation Function: The output of the convolutional layer is passed through a non-linear transformation using an activation function, commonly ReLU (Rectified Linear Unit), to introduce non-linearity.

Pooling Layer: This layer performs down sampling on the feature maps, reducing their dimensions while retaining important features.

Max pooling and average pooling are two common types of pooling used in deep learning models as shown in Fig. 10. Max pooling is a pooling operation that selects the maximum value from a specific region of the input data. Average pooling, on the other hand, calculates the average value of a specific region of the input data [88].

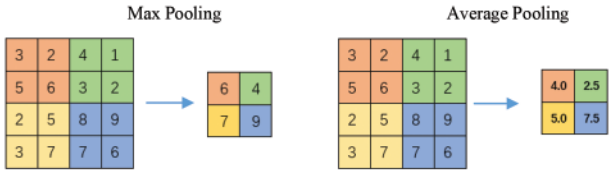

Figure 10: The pooling operation (max pooling and average pooling)

Fully Connected Layer: The output of the pooling layer is connected to a fully connected neural network, where feature fusion and classification are performed.

Output Layer: Depending on the task type, the output layer can consist of one or multiple neurons, used for tasks such as classification, detection, or segmentation.

The structure diagram of CNNs can be adjusted and expanded based on specific network architectures and task requirements. The provided diagram is a basic representation. In practical applications, additional layers such as batch normalization and dropout can be added to improve the model's performance and robustness.

5.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a specific type of artificial neural network designed to effectively process sequential data. They demonstrate exceptional performance in tasks involving time-dependent or sequential information, such as natural language processing, speech recognition, and time series analysis. RNN also plays a crucial role in SLR. In the following sections, we will introduce the fundamental concepts of RNN, as well as their various variants, including Long Short-Term Memory Networks (LSTM), Gate Recurrent Units (GRU), and Bidirectional Recurrent Neural Networks (BiRNN).

5.2.1 Basics of RNNs

One of the main advantages of RNNs is their ability to handle variable-length input sequences. This makes them well-suited for tasks such as sentiment analysis, where the length of the input text can vary. Additionally, RNNs can capture long-term dependencies in the data, allowing them to model complex relationships over time [89]. Fig. 11 displays the classic structure of a RNNs and its unfolded state.

The left side of Fig. 11 represents a classic structure of a Recurrent Neural Network (RNN), where X represents the input layer, S represents the hidden layer, and O represents the output layer. U, V, and W are parameters in the network. The hidden state represented by S is not only used to pass information to the output layer but also fed back to the network through the arrow loop. Unfolding

this recurrent structure results in the structure shown on the right side of Fig. 11: X_{t-1} to X_{t+1} represent sequentially input data at different time steps, and each time step's input generates a corresponding hidden state S. This hidden state at each time step is used not only to produce the output at that time step but also participates in calculating the next time step's hidden state.

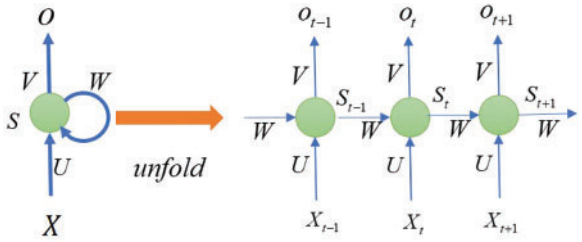

Figure 11: A classic structure of RNN and its unfolded state

RNNs are neural networks that excel at processing sequential data. However, they suffer from the vanishing or exploding gradient problem, which hinders learning. To address this, variants like Long Short-Term Memory Networks (LSTM) and Gate Recurrent Units (GRU) have been developed, incorporating gating mechanisms to control information flow.

5.2.2 Long Short-Term Memory Networks (LSTM)

LSTM [90] are a type of RNN that have gained significant popularity in the field of deep learning. LSTM is specifically designed to address the vanishing gradient problem, which is a common issue in training traditional RNNs.

The key characteristic of LSTM networks is their ability to capture long-term dependencies in sequential data. This is achieved through the use of memory cells that can store and update information over time. Each memory cell consists of three main components: an input gate, a forget gate, and an output gate [89], as shown in Fig. 12.

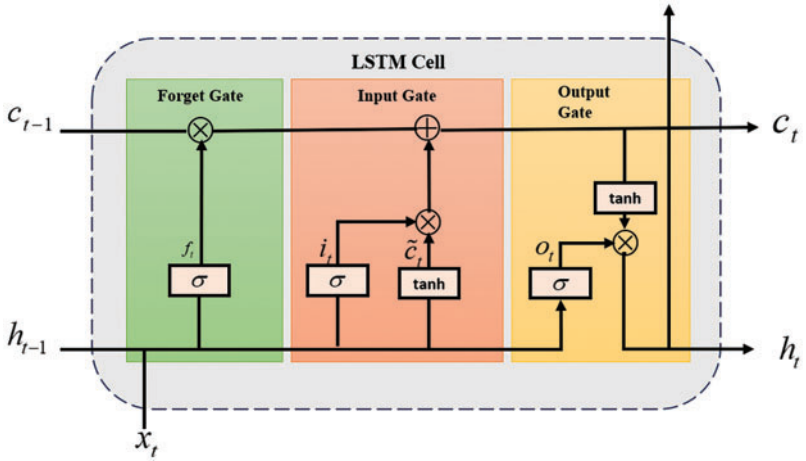

Figure 12: Typical structure of LSTM

The input gate determines how much new information should be added to the memory cell. It takes into account the current input and the previous output to make this decision. The forget gate

controls the amount of information that should be discarded from the memory cell. It considers the current input and the previous output as well. Finally, the output gate determines how much of the memory cell's content should be outputted to the next step in the sequence. By using these gates, LSTM networks can selectively remember or forget information at each time step, allowing them to capture long-term dependencies in the data. This is particularly useful in tasks such as speech recognition, machine translation, and sentiment analysis, where understanding the context of the entire sequence is crucial.

Another important characteristic of LSTM is its ability to handle gradient flow during training. The vanishing gradient problem occurs when gradients become extremely small as they propagate backward through time in traditional RNNs. LSTM addresses this issue by using a constant error carousel, which allows gradients to flow more freely and prevents them from vanishing or exploding.

5.2.3 Gate Recurrent Unit (GRU)

Similar to LSTM, the GRU is a RNN architecture that has gained significant attention in the field of natural language processing and sequence modeling [91,92]. It shares similarities with LSTM in terms of its ability to capture long-term dependencies in sequential data and mitigate the vanishing gradient problem encountered in traditional RNN.

One key feature of the GRU is its utilization of two gates: the update gate and the reset gate. These gates play a crucial role in controlling the flow of information within the network. The update gate determines how much of the previous hidden state should be retained and how much new information from the current input should be incorporated into the next hidden state. This allows the GRU to selectively update its memory and capture relevant information over long sequences.

The reset gate, on the other hand, determines how much of the previous hidden state should be forgotten or reset. By resetting a portion of the hidden state, the GRU can adaptively discard irrelevant information and focus on capturing new patterns in the input sequence.

In addition to its gating mechanism, the GRU offers certain advantages over LSTM in terms of computational efficiency and simplicity. With a reduced number of gates, the GRU requires fewer computational resources and is easier to train, especially when dealing with smaller datasets. The GRU achieves this efficiency while still maintaining its ability to selectively update its memory and control the flow of information.

Furthermore, the GRU is flexible and adaptable, making it suitable for various tasks and datasets. Researchers have proposed variations of the GRU, such as the Gated Feedback Recurrent Neural Network (GF-RNN), which further enhance its memory capacity and extend its capabilities.

5.2.4 Bidirectional RNN(BiRNN)

Bidirectional Recurrent Neural Networks (BiRNN) [93] have gained significant attention in the field of natural language processing and sequential data analysis. The unique characteristic of BiRNN lies in their ability to capture both past and future context information in sequential data. This is achieved through the use of two recurrent neural networks, one processing the input sequence in the forward direction and the other in the reverse direction. By combining the outputs of both networks, BiRNN can provide a more comprehensive understanding of the sequential data [94]. The structure of a BiRNN is shown in Fig. 13.

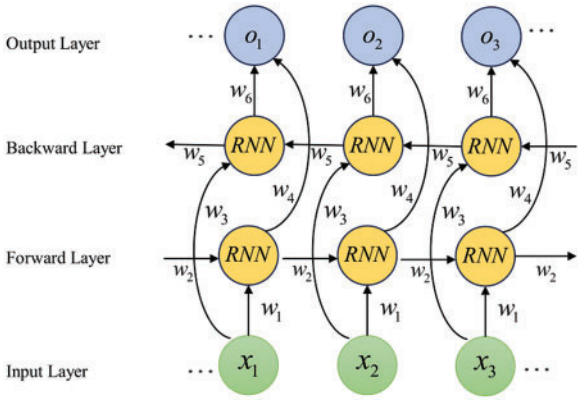

Figure 13: The structure of BiRNN

In terms of training, BiRNN typically employs backpropagation through time (BPTT) or gradient descent algorithms to optimize the network parameters. However, the bidirectional nature of BiRNN introduces challenges in training, as information from both directions needs to be synchronized. To address this issue, techniques such as sequence padding and masking are commonly used. The basic unit of a BiRNN can be a standard RNN, as well as a GRU or LSTM unit. In practice, for many Natural Language Processing (NLP) problems involving text, the most used type of bidirectional RNN model is the one with LSTM units.

5.3 Deep Transfer Learning

Transfer learning, initially proposed by Tom Mitchell in 1997, is a technique that aims to transfer knowledge and representations from one task or domain to another, thereby improving performance [95]. In recent years, deep transfer learning has emerged as a prominent approach that leverages deep neural networks for effective knowledge transfer [96–98].The following will introduce the pre-training model commonly used in the field of sign language recognition.

5.3.1 VGGNet

VGGNet [99] was developed by the Visual Geometry Group at the University of Oxford and has achieved remarkable performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). One of VGGNet's notable characteristics is its uniform architecture. It comprises multiple stacked convolutional layers, allowing for deeper networks with 11 to 19 layers, enabling the network to learn intricate features and patterns.

Among the variants of VGGNet, VGG-16 is particularly popular. As shown in Fig. 14, VGG-16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. It utilizes 3×3 convolutional filters, followed by max-pooling layers with a 2×2 window and stride of 2. The number of filters gradually increases from 64 to 512. The network also incorporates three fully connected layers with 4096 units each, employing a ReLU activation function. The final output layer consists of 1000 units representing the classes in the ImageNet dataset, utilizing a softmax activation function.

While VGGNet has demonstrated success, it has limitations. The deep architecture of VGGNet results in computationally expensive and memory-intensive operations, demanding substantial computational resources. However, the transfer learning capability of VGGNet is a significant advantage. With pre-trained VGG models, which have been trained on large datasets such as ImageNet,

researchers and practitioners can conveniently utilize them as a starting point for other computer vision tasks. This has greatly facilitated research and development in the field.

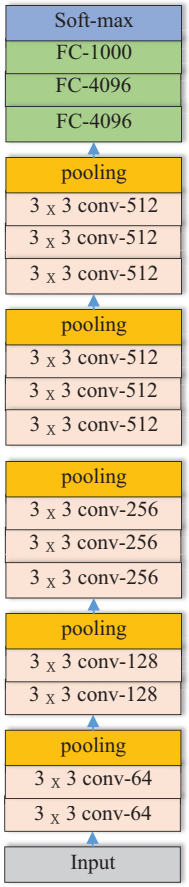

Figure 14: Architecture of VGG-16

5.3.2 GoogLeNet (Inception)

GoogLeNet, developed by a team of researchers at Google led by Christian Szegedy, aimed to create a more efficient and accurate Convolutional Neural Network (CNN) architecture for image classification tasks [100].

One of the key innovations introduced by GoogLeNet is the Inception module, as shown in Fig. 15 This module utilizes multiple parallel convolutional layers with different filter sizes, including 1×1, 3×3, and 5×5 filters. This allows the network to capture features at various scales, enhancing its ability to recognize complex patterns. The parallel layers within the Inception module capture features at different receptive field sizes, enabling the network to capture both local and global information. Additionally, 1×1 convolutions are employed to compute reductions before the more computationally expensive 3×3 and 3×3 convolutions. These 1×1 convolutions serve a dual purpose, not only reducing the dimensionality but also incorporating rectified linear activation.

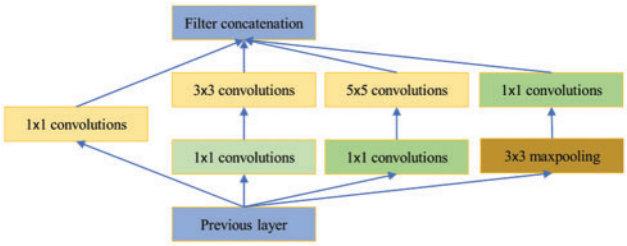

Figure 15: Inception module with dimension reductions

Over time, the Inception of GoogLeNet evolved with versions like Inception-v2 [101], Inception-v3 [102], Inception-v4 [103], and Inception-ResNet [103]. These versions introduced various enhancements, including batch normalization [101], optimized intermediate layers, label smoothing, and the combination of Inception with ResNet's residual connections. These improvements led to higher accuracy, faster convergence, and better computational efficiency.

5.3.3 ResNet

ResNet, short for Residual Network, was introduced by Kaiming He and his team from Microsoft Research in 2015 [104]. It was specifically designed to address the problem of degradation in very deep neural networks.

Traditional deep neural networks face challenges in effectively learning transformations as they become deeper. This is due to the vanishing or exploding gradients during backpropagation, making it difficult to optimize the weights of deep layers. ResNet tackles this issue by introducing residual connections, which learn the residual mapping—the difference between the input and output of a layer [104]. The architecture of residual connections is illustrated in Fig. 16. The input passes through convolutional layers and residual blocks. Each residual block contains multiple convolutional layers, with the input added to the block's output through a skip connection. This allows the gradient to flow directly to earlier layers, addressing the vanishing gradient problem.

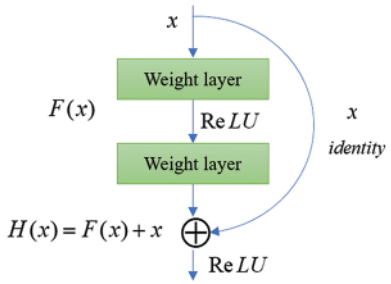

Figure 16: The architecture of residual connections

Mathematically, the residual connection is represented as $H(x) = F(x) + x$. Here, x is the input to a layer, $F(x)$ is the layer's transformation, and $H(x)$ is the output. The residual connection adds the input x to the transformed output F(x), creating the residual mapping $H(x)$. The network learns to optimize this mapping during training. ResNet's architecture has inspired the development of other residual-based models like ResNeXt [105], Wide ResNet [106], and DenseNet [107], which have further improved performance in various domains.

5.3.4 Lightweight Networks (MobileNet)

Traditional CNN architectures are often computationally intensive and memory-consuming, posing challenges for deployment on resource-constrained devices. To address these challenges, Lightweight Networks have emerged as a solution. These networks are specifically designed to be compact and efficient, enabling their deployment on devices with limited computational resources. Lightweight Networks achieve this by employing various strategies such as reducing the number of parameters, employing efficient convolutional operations, and optimizing network architectures. Several specific models have been developed in the realm of Lightweight Networks like MobileNet [108], ShuffleNet [109], SqueezeNet [110], and EfficientNet [111]. One of the most prominent and widely used models is MobileNet.

MobileNet is designed to strike a balance between model size and accuracy. It adopts depthwise separable convolutions to significantly reduce the computational cost compared to standard convolutional layers. This approach decomposes the convolution operation into a depthwise convolution followed by a pointwise convolution, effectively reducing the number of operations required. Fig. 17 compares a layer that uses regular convolutions, batch normalization, and ReLU nonlinearity with a layer that utilizes depthwise convolution [112], followed by a 1×1 pointwise convolution, along with batch normalization and ReLU after each convolutional layer. The two layers are different in terms of their architectural design and the operations performed at each step, as shown in Fig. 17b.

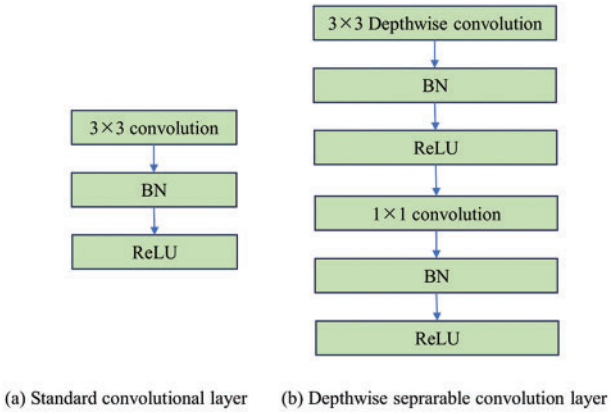

Figure 17: Comparison between standard convolutional layer and depthwise seprarable convolutions

MobileNet has multiple variations, including MobileNetV1 [108], MobileNetV2 [113], and MobileNetV3 [114]. Each version improves upon the previous one by introducing new techniques to further enhance efficiency and accuracy. MobileNetV2, for example, introduces inverted residual blocks and linear bottleneck layers to achieve better performance. MobileNetV3 leverages a combination of channel and spatial attention modules to improve both speed and accuracy.

5.3.5 Transformer

The transformer model, proposed by Vaswani et al. in 2017, has emerged as a breakthrough in the field of deep learning. Its self-attention mechanism, parallelizable computation, ability to handle variable-length sequences, and interpretability have propelled it to the forefront of research in natural language processing and computer vision [115].

The Transformer model architecture consists of two main components: the encoder and the decoder. These components are composed of multiple layers of self-attention and feed-forward neural networks, as shown in Fig. 18.

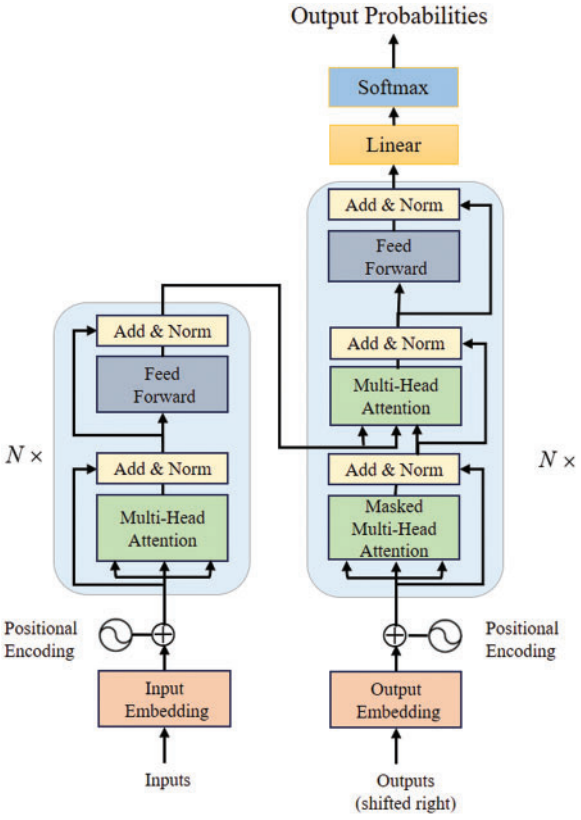

Figure 18: The transformer-model architecture

The encoder takes an input sequence and processes it to obtain a representation that captures the contextual information of each element in the sequence. The input sequence is first embedded into a continuous representation, which is then passed through a stack of identical encoder layers.

Each encoder layer in the Transformer model architecture has two sub-layers: a multi-head self-attention mechanism and a feed-forward neural network. The self-attention mechanism allows the model to attend to different parts of the input sequence when processing each element, capturing the relationships and dependencies between elements. The feed-forward neural network applies a non-linear transformation to each element independently, enhancing the model's ability to capture complex patterns in the data.

The decoder, on the other hand, generates an output sequence based on the representation obtained from the encoder. It also consists of a stack of identical layers, but with an additional sub-layer that performs multi-head attention over the encoder's output. This allows the decoder to focus on relevant parts of the input sequence when generating each element of the output sequence.

In addition to the self-attention mechanism, the Transformer model architecture incorporates positional encodings to handle the order of elements in the input sequence. These positional encodings

are added to the input embeddings, providing the model with information about the relative positions of elements. This enables the model to differentiate between different positions in the sequence.

The Transformer model architecture is trained using a variant of the attention mechanism called "scaled dot-product attention". This mechanism computes the attention weights between elements in the sequence by taking the dot product of their representations and scaling the result by the square root of the dimension of the representations. The attention weights are then used to compute a weighted sum of the representations, which forms the output of the attention mechanism.

The impact of the Transformer architecture is evident in its state-of-the-art performance across various domains, establishing it as a fundamental building block in modern deep learning models. These new models, including GPT [116], BERT [117], T5, ViT [118], and DeiT, are all based on the Transformer architecture and have achieved remarkable performance in a wide range of tasks in natural language processing and computer vision, making significant contributions to these fields.

6 Analysis and Discussion

In this section, we have organized the techniques and methods of SLR based on deep learning, focusing on fingerspelling, isolated words, and continuous sign language. To ensure our readers stay abreast of the latest advancements in SLR, we have placed a strong emphasis on recent progress in the field, primarily drawing from cutting-edge research papers published within the past five years.

To help understand more clearly, we provided a table with all abbreviations and full names as follows in Table 4.

Table 4: List of all abbreviations and full names

Abbreviations	Full names
AE	Autoencoder
AgSL	Argentinian sign language
ArSL	Arabic sign language
ARS-MA	Accuracy-based weighted voting
ASL	American sign language
AsSL	Assamese sign language
BdSL	Bangla sign language
Bi-LSTM	Bi-directional long short term memory
BN	Batch normalization
BSL	British sign language
BrSL	Brazilian sign language
CA	Contrastive attention
CAE	convolutional autoencoders
CNN	Convolutional neural network
CSL	Chinses sign language
CSLR	Continuous sign language recognition
CSOM	Convolutional self-organizing map
CTC	Connectionist temporal classification
D	Depth

(Continued)

Table 4 (continued)

Abbreviations	Full names
DA	Data augmentation
DBN	Deep belief net
DR	Dropout techniques
DRL	Deep reinforcement learning
GSL	German sign language
H-GANs	Hyperparameter based optimized generative adversarial networks
HP	Hand pose
IMU	Inertial measurement unit
IMUs	Inertial measurement units
ISL	Indian sign language
JSL	Japanese sign language
KD	Knowledge distillation
KSU-ArSL	King Saud University Arabic sign language dataset
LRN	Local response normalization
LSE	Spanish sign language
LSTM	Long short-term memory
MC-LSTMs	Multi-cue long short-term memory networks
MHA	Multi-head attention
MoSL	Moroccan sign language
PSL	Persian sign language
ReLU	Rectified linear unit
RKD	random knowledge distillation strategy
RSL	Russian sign language
RST	Relative sign transformer
RTS	Rotated, translated and scaled
SA	Statistical attention
OF	Optical flow
SaSL	Saudi sign language
sEMG	surface electromyography
SF	Scene flow
SLVM	Sign language video in museums dataset
SMKD	Self-mutual knowledge distillation
SP	Stochastic pooling
SSD	Single shot detector
STFE-Net	Spatial-temporal feature extraction network
ST-GCNs	Spatial-temporal graph convolutional networks

(Continued)

Table 4 (continued)

Abbreviations	Full names
STMC	Spatial-temporal multi-cue
SVAE	Stacked variational auto-encoders
PHOENIX	RWTH-PHOENIX Weather
PHOENIX14	RWTH-PHOENIX Weather-2014
PHOENIX14T	RWTH-PHOENIX Weather-2014-T
TaSL	Tactical sign language
TFSL	Thai finger-spelling sign language
TuSL	Turkish sign language
ViT	Vision transformer
WER	Word error rate
WLASL	word-level American sign language
ZSL	Zero-shot learning

6.1 Fingerspelling Recognition

In recent years, numerous recognition techniques have been proposed in the literature, with a specific focus on utilizing deep learning methods to recognize fingerspelling symbols. Table 5 provides a summary of various example fingerspelling recognition systems.

Table 5: Summary of various example fingerspelling recognition systems (Pr: Precision)

Paper	Year	Language	Modality	Methods	Performance (Acc.)
[119]	2019	CSL	Image (RGB)	CNN (BN+leaky ReLU+DT+DR)	88.10 +/− 1.48%(Pr)
[120]	2019	ASL	Image(D)	PCANet+SVM	88.70%
[121]	2020	CSL	Image (RGB)	DA+CNN (BN+DR+SP)	89.32 +/− 1.07%(Pr)
[122]	2021	BdSL	Image (RGB)	Transfer learning+ZSL	93.68%
[123]	2021	LSE	Image (RGB)	Deep transfer learning (VGG-16, ResNet, EfficientNet)	96.42%
[124]	2021	TSL	Image (RGB)	TSLNet (CNN+BN+DR+MaxPooling)	99.7%
[125]	2021	TFSL	Video+OF	AlexNet	88.00% (one stroke)
[126]	2021	ASL	Image (RGB) (training)+ Real-Time (test)	YOLO+LSTM	98.07%
[56]	2022	CSL	Image (RGB)	CNN (Three different combinations of blocks: Conv-BN-ReLU-Pooling, Conv-BN-ReLU, Conv-BN-ReLU-BN)	94.88 +/− 0.99%
[127]	2022	BdSL	Image (RGB)	transfer learning (ResNet18)	99.99%

(Continued)

Table 5 (continued)

Paper	Year	Language	Modality	Methods	Performance (Acc.)
[128]	2022	ASL	Image (RGB)	ensemble learning (feature extraction: LeNet, AlexNet, VGGNet, GoogleNet, and ResNet+ classification: ARS-MA)	98.83%
[129]	2022	CSL	data glove	CNN	99.50%
[130]	2022	ASL	IMU Senor	feature extraction: time/time-frequency domain/angle-based features classification: CTC recognition: encoder-decoder	Nearly 100% (Within-User) 74.8% (Cross-User)
[131]	2022	ASL	Image (RGB)	CNN	99.38%
[132]	2023	BdSL	Image (RGB)	deep transfer learning + random forest classifier	91.67%
[133]	2023	ASL	Image (RGB)	MobileNetV2	98.77%
[134]	2023	ArSL	Image (RGB)	MobileNet	94.46%
[135]	2023	AsSL	Image (RGB)	MediaPipe	99%
[136]	2023	ISL	Image (RGB)	Transformer	99.29
[137]	2023	ISL	Image (RGB)	CNN (data augmentation, BN, dropout, stochastic pooling, diffGrad optimizer)	99.76%
[138]	2023	ASL BdSL	Image (RGB)	Attention + MobileNetV2	99.95% 92.1%

6.1.1 Analysis of Fingerspelling Recognition

As shown in Table 5, CNN has emerged as the dominant approach in fingerspelling recognition technology, demonstrating excellent performance. Jiang et al. employed CNN on fingerspelling recognition of CSL. In 2019, they proposed a six-layer deep convolutional neural network with batch normalization, leaky ReLU, and dropout techniques. Experimental results showed that the approach achieved an overall accuracy of 88.10% +/− 1.48% [119]. In 2020, they further proposed an eight-layer convolutional neural network, incorporating advanced techniques such as batch normalization, dropout, and stochastic pooling. Their method achieved the highest accuracy of 90.91% and an overall accuracy of 89.32% +/− 1.07% [121]. Their latest contribution in 2022 is the CNN-CB method, which employed three different combinations of blocks: Conv-BN-ReLU-Pooling, Conv-BN-ReLU, and Conv-BN-ReLU-BN. This new method achieved an outstanding overall accuracy of 94.88% +/− 0.99% [56]. Martinez-Martin et al. [123] introduced a system for recognizing the Spanish sign language alphabet. Both CNN and RNN were tested and compared. The results showed that CNN achieved significantly higher accuracy, with a maximum value of 96.42%. Zhang et al. [129] presented a real-time system for recognizing alphabets, which integrated visual data with a data glove equipped with flexible strain sensors and somatosensory data from a camera. The system utilized CNN to fuse and recognize the sensor data. The proposed system achieved a high recognition rate of up to 99.50% for all 26 letters. Kasapbaşi et al. [131] established a new dataset of the American Sign Language alphabet and developed a fingerspelling recognition system using CNN. The experiments showed that the accuracy reached 99.38%.

In recent years, there has been rapid development in the field of deep transfer learning and ensemble learning, and a set of pre-trained models has been applied to fingerspelling recognition. Sandler et al. [113] introduced two methods for automatic recognition of the BdSL alphabet, utilizing conventional transfer learning and contemporary zero-shot learning (ZSL) to identify both seen and unseen data. Through extensive quantitative experiments on 18 CNN architectures and 21 classifiers, the pre-trained DenseNet201 architecture demonstrated exceptional performance as a feature extractor. The top-performing classifier, identified as Linear Discriminant Analysis, achieved an impressive overall accuracy of 93.68% on the extensive dataset used in the study. Podder et al. [127] compared the classification performance with and without background images to determine the optimal working model for BdSL alphabet classification. Three pre-trained CNN models, namely ResNet18 [104], MobileNet_V2 [113], and EfficientNet_B1 [111], were used for classification. It was found that ResNet18 achieved the highest accuracy of 99.99%. Ma et al. [128] proposed an ASL recognition system based on ensemble learning, utilizing multiple pre-trained CNN models including LeNet, AlexNet, VGGNet, GoogleNet, and ResNet for feature extraction. The system incorporated accuracy-based weighted voting (ARS-MA) to improve the recognition performance. Das et al. [132] proposed a hybrid model combining a deep transfer learning-based CNN with a random forest classifier for automatic recognition of BdSL alphabet.

Some models have combined two or more approaches in order to boost the recognition accuracy. Aly et al. [120] presented a novel user-independent recognition system for the ASL alphabet. This system utilized the PCANet, a principal component analysis network, to extract features from depth images captured by the Microsoft Kinect depth sensor. The extracted features were then classified using a linear support vector machine (SVM) classifier. Rivera-Acosta et al. [126] proposed a novel approach to address the accuracy loss when training models to interpret completely unseen data. The model presented in this paper consists of two primary data processing stages. In the first stage, YOLO was employed for handshape segmentation and classification. In the second stage, a Bi-LSTM was incorporated to enhance the system with spelling correction functionality, thereby increasing the robustness of completely unseen data.

Some SLR works have been deployed in embedded systems and edge devices, such as mobile devices, Raspberry Pi, and Nareshkumar et al. [133] utilized MobileNetV2 on terminal devices to achieve fast and accurate recognition of letters in ASL, reaching an accuracy of 98.77%. MobileNet was utilized to develop a model for recognizing the Arabic language's alphabet signs, with a recognition accuracy of 94.46% [134]. Zhang et al. [138]introduced a novel lightweight network model for alphabet recognition, incorporating an attention mechanism. Experimental results on the ASL dataset and BdSL dataset demonstrated that the proposed model outperformed existing methods in terms of performance. Ang et al. [139] implemented a fingerspelling recognition model for Filipino Sign Language using Raspberry Pi. They used YOLO-Lite for hand detection and MobileNetV2 for classification, achieving an average accuracy of 93.29% in differentiating 26 hand gestures representing FSL letters. Siddique et al. [140] developed an automatic Bangla sign language (BSL) detection system using deep learning approaches and a Jetson Nano edge device.

6.1.2 Discussion about Fingerspelling Recognition

According to the data in Table 4, fingerspelling recognition has achieved impressive results, with the majority of models achieving accuracy rates of 90% or higher. This high performance can be attributed to several factors.

Firstly, fingerspelling recognition focuses on a specific set of alphabets and numbers. For instance, in Chinese fingerspelling sign language, there are 30 sign language letters, including 26 single letters (A to Z) and four double letters (ZH, CH, SH, and NG) [141]. This limited vocabulary simplifies the training process and contributes to higher accuracy rates.

Secondly, fingerspelling recognition primarily deals with static images, allowing it to concentrate on recognizing the hand configurations and positions associated with each sign. As a result, there is no need to consider continuous motion, which further enhances accuracy.

Thirdly, sign language databases are typically captured in controlled environments, free from complex lighting and background interference. This controlled setting reduces the complexity of the recognition task, leading to improved accuracy.

While fingerspelling recognition has achieved remarkable results due to the limited vocabulary and clear visual cues of static signs, there are still areas that can be improved. These include addressing the variability in hand shapes among handling variations in lighting and background conditions, and the development of real-time recognition systems.

6.2 Isolated Sign Language Recognition

Isolated sign language recognition (isolated SLR) refers to the task of recognizing individual sign language gestures or signs in a discrete manner. It focuses solely on recognizing and classifying isolated signs without considering the temporal relationship between them. In this approach, each sign is treated as an independent unit, and the recognition system aims to identify the meaning of each individual sign. Table 6 lists a number of proposed approaches for isolated SLR.

Table 6: Summary of recognition systems for isolated signs

Paper	Year	Language	Modalities	Database (categories/samples/signers)	Methods	Component	Accuracy
[10]	2018	CSL	Infrared + Contour+ Skeleton	SLVM (20/6800/17) (own)	3DCNN+Multi-modal fusion	H	89.2%
[61]	2019	CSL	RGB + D + Skeleton	500/125000/50 (own)	3DCNN+Attention	H	88.7%
[142]	2020	ASL	sensor	27/38451/12 (own)	RNN (LSTM)	H	99.81%
[143]	2020	ArSL	RGB	23/-/3 (own)	DeepLabv3++CSOM +BiLSTM	H	89.59%
[144]	2020	CSL	sEMG + ACC + GYRO	150/-/8 (own)	DBN	H	95.1% (user-dependent) 88.2% (user-independent)
[145]	2020	PSL	RGB	100/10000/10 (own)	SSD+CNN (ResNet50)+ LSTM	H+HP	98.42%
[146]	2021	ISL	RGB	13/-/- (own)	GoogleNet+BiLSTM	H	76.21%
[147]	2021	TSL	Skeleton	KLEF3DSL_2Dskeletal (200/15000/15)	Multiview CNN	H	–
[148]	2021	SaSL	RGB	KSU-ArSL	Inception+BiLSTM	H	85.6%
[13]	2021	ASL	RGB	Boston ASL (Lexicon Video Dataset)(100/3300/6)	Cascaded 3DCNN	H	96% (Pr)
[149]	2021	ISL	RGB + D RGB + D RGB + D RGB + D	Montalbano II (20/ 12,575/) isoGD(249/47,933/21) MSR Daily Activity 3D (16/320/-) CAD-60 (12/60/-)	fine-tuned AlexNet+LSTM	H +HP	99.08% 86.10% 98.40% 95.50%
[150]	2022	ArSL	RGB	56/224/5 (own)	2DCRNN+3DCNN	H	92% (2DCRNN) 99% (3DCNN)
[86]	2022	MoSL	RGB	497/ 2171/- (own)	3D GS-Net (AE+3DCNN)	H +F	99.18 (Pr)
[37]	2022	ASL	RF Sensor	20/-/- (own)	Transfer learning+CAE	H	95%
[151]	2022	ISL	RGB	IISL2020 (11/12100/16)	InceptionResNetV2+LSTM+GRU	H	97%
[152]	2022	TSL	RGB	32/3200/3 (own)	DA+transfer learning (VGG166/Inception/ResNet)+SVM	H	–
[153]	2022	TSL	RGB	30/-/- (own)	TCNN+BiLSTM+ knowledge distillation	H	92.67%
[154]	2023	ISL	RGB	COVID-19-related words (17/340/5)	VGG16+BiLSTM	H	83.36%
[155]	2023	ArSL	RGB	20/8467/20 (own)	CNN+RNN	H	98%
[156]	2023	ASL AgSL BrSL	RGB RGB RGB+D+Skeleton	DSL-46 (46/2910/) LSA64 (64/3200/10) LIBRAS-BSL (37/4440/10)	MediaPipe+GRU+ 1DCNN	H	98.8%/ 99.84%/ 88.40%
[157]	2023	ISL	Sensor	50/-/- (own)	CNN+Deep Ensemble (bagging+soft voting)	H	96.6%
[158]	2023	ISL	RGB	INCLUDE (263/4287/-)	VGG-19+BiLSTM	H	87.67%
[159]	2023	Turkish SL	RGB, D, and skeleton data (kinect)	BosphorusSign22k AUTSL	ST-GCN+MC-LSTM	H+F+B	–

(Continued)

Table 6 (continued)

Paper	Year	Language	Modalities	Database (categories/samples/signers)	Methods	Component	Accuracy
[160]	2023	JSL	RGB RF Sensor	Twenty Sign Word Dataset (20/18000/-) Five Sign Word Dataset (5/231/-)	RTS+Hybrid Segmentation+CNN	H	99.10% 98.00%
[161]	2023	ASL	RGB	WLASL100 WLASL300 WLASL1000 WLASL2000	Sign2Pose: YOLOv3+Transformer	HP	80.9% 64.21% 49.46% 38.65%
[162]	2023	ISL RSL	RGB RGB	ISLW(500-2500-7)(own) RSL(1100-37775-3)(own)	static gesture recognition:3D-CNN dynamic gesture recognition: semantic spatial multi-cue feature, modified auto-encoder	H +F+B	99.76% (static) 99.85% (dynamic)

Note: H: Hand; HP: Hand Pose; F: Face; B: Body; Pr: Precision.

6.2.1 3DCNN-Based Approach for Isolated SLR

3DCNN can analyze video data directly, incorporating the temporal dimension into the feature extraction process. This allows 3DCNN to capture motion and temporal patterns, making them suitable for isolated SLR.

Liang et al. [10] presented a data-driven system that utilizes 3DCNN for extracting both spatial and temporal features from video streams. The motion information was captured by observing the depth variation between consecutive frames. Additionally, multiple video streams, such as infrared, contour, and skeleton, are utilized to further enhance performance. The proposed approach was evaluated on the SLVM dataset, a multi-modal dynamic sign language dataset captured with Kinect sensors. The experimental results demonstrated an accuracy improvement of 89.2%. Huang et al. [61] introduced attention-based 3DCNN for isolated SLR. The framework offered two key advantages: Firstly, 3DCNN was capable of learning spatiotemporal features directly from raw video data, without requiring prior knowledge. Secondly, the attention mechanism incorporated in the network helped to focus on the most relevant clues. Sharma et al. [13] utilized 3DCNN, which was effective in identifying patterns in volumetric data such as videos. The cascaded 3DCNN was trained using the Boston ASL (Lexicon Video Dataset) LVD dataset. The proposed approach surpassed the current state-of-the-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%). Boukdir et al. [138] proposed a novel approach based on a deep learning architecture for classifying Arabic sign language video sequences. This approach utilized two classification methods: 2D Convolutional Recurrent Neural Network (2DCRNN) and 3D Convolutional Neural Network (3DCNN). In the first method, the 2DCRNN model was employed to extract features with a recurrent network pattern, enabling the detection of relationships between frames. The second method employed the 3DCNN model to learn spatiotemporal features from smaller blocks. Once features were extracted by the 2DCRNN and 3DCNN models, a fully connected network was utilized to classify the video data. Through four-fold cross-validation, the results demonstrated a horizontal accuracy of 92% for the 2DCRNN model and 99% for the 3DCNN model.

6.2.2 CNN-RNN Hybrid Models for Isolated SLR

The CNN-RNN (LSTM or GRU) hybrid model offers a powerful framework for isolated SLR by leveraging the spatial and temporal information in sign language videos.

Rastgoo et al. [145] introduced an efficient cascaded model for isolated SLR that leverages spatio-temporal hand-based information through deep learning techniques. Specifically, the model incorporated the use of Single Shot Detector (SSD), CNN, and LSTM to analyze sign language videos. Venugopalan et al. [146] presented a hybrid deep learning model that combined a convolutional LSTM network for the classification of ISL. The proposed model achieved an average classification accuracy of 76.21% on the ISL agricultural word dataset. Rastgoo et al. [149] introduced a hand pose-aware model for recognizing isolated SLR using deep learning techniques. They proposed various models, incorporating different combinations of pre-trained CNN models and RNN models. In their final model, they utilized the AlexNet and LSTM for hand detection and hand pose estimation. Through experimental evaluation, they achieved notable improvements in accuracy, with relative improvements of 1.64%, 6.5%, and 7.6% on the Montalbano II, MSR Daily Activity 3D, and CAD-60 datasets, respectively. Das et al. [158] proposed a vision-based SLR system called Hybrid CNN-BiLSTM SLR (HCBSLR). The HCBSLR system addressed the issue of excessive pre-processing by introducing a Histogram Difference (HD) based key-frame extraction method. This method improved the accuracy and efficiency of the system by eliminating redundant or useless frames. The HCBSLR system

utilized VGG-19 for spatial feature extraction and employed BiLSTM for temporal feature extraction. Experimental results demonstrated that the proposed HCBSLR system achieved an average accuracy of 87.67%.

Due to limited storage and computing capacities on mobile phones, the implementation of SLR applications is often restricted. To address this issue, Abdallah et al. [156] proposed the use of lightweight deep neural networks with advanced processing for real-time dynamic sign language recognition (DSLR). The application leveraged two robust deep learning models, namely the GRU and the 1D CNN, in conjunction with the MediaPipe framework. Experimental results demonstrated that the proposed solution could achieve extremely fast and accurate recognition of dynamic signs, even in real-time detection scenarios. The DSLR application achieved high accuracies of 98.8%, 99.84%, and 88.40% on the DSL-46, LSA64, and LIBRAS-BSL datasets, respectively. Li et al. [153] presented MyoTac, a user-independent real-time tactical sign language classification system. The network was made lightweight through knowledge distillation by designing tactical CNN and BiLSTM to capture spatial and temporal features of the signals. Soft targets were extracted using knowledge distillation to compress the neural network scale nearly four times without affecting the accuracy.

Most studies on SLR have traditionally focused on manual features extracted from the shape of the dominant hand or the entire frame. However, it is important to consider facial expressions and body gestures. Shaik et al. [147] proposed an isolated SLR framework that utilized Spatial-Temporal Graph Convolutional Networks (ST-GCNs) [151,152] and Multi-Cue Long Short-Term Memories (MC-LSTMs) to leverage multi-articulatory information (such as body, hands, and face) for recognizing sign glosses.

6.2.3 Sensor—DNN Approaches for Isolated SLR

The advantages of sensor and DNN make sensor-DNN approaches a promising choice for effective and practical isolated SLR systems. Lee et al. [142] developed and deployed a smart wearable system for interpreting ASL using the RNN-LSTM classifier. The system incorporated sensor fusion by combining data from six IMUs. The results of the study demonstrated that this model achieved an impressive average recognition rate of 99.81% for 27 word-based ASL. In contrast to video, RF sensors offer a way to recognize ASL in the background without compromising the privacy of Deaf signers. Gurbuz et al. [37] explored the necessary RF transmit waveform parameters for accurately measuring ASL signs and their impact on word-level classification accuracy using transfer learning and convolutional autoencoders (CAE). To improve the recognition accuracy of fluent ASL signing, a multi-frequency fusion network was proposed to utilize data from all sensors in an RF sensor network. The use of the multi-frequency fusion network significantly increased the accuracy of fluent ASL recognition, achieving a 95% accuracy for 20-sign fluent ASL recognition, surpassing conventional feature-level fusion by 12%. Gupta et al. [157] proposed A novel ensemble of convolution neural networks (CNN) for robust ISL recognition using multi-sensor data.

6.2.4 Other Approaches for Isolated SLR

Some researchers used Deep belief net (DBN), Transformer, and other models for isolated SLR. Aly et al. [143] proposed A novel framework for recognizing ArSL that is not dependent on the signer. This framework utilized multiple deep learning architectures, including hand semantic segmentation, hand shape feature representation, and deep recurrent neural networks. The DeepLabv3+ model was employed for semantic segmentation. Handshape features were extracted using a single layer Convolutional Self-Organizing Map (CSOM). The extracted feature vectors were then recognized

using a deep BiLSTM. Deep belief net (DBN) was applied to the field of wearable-sensor-based CSL recognition [144].To obtain multi-view deep features for recognition, Shaik et al. [147] proposed using an end-to-end trainable multi-stream CNN with late feature fusion. The fused multi-view features are then fed into a two-layer dense network and a softmax layer for decision-making. Eunice et al. [161] proposed a novel approach for gloss prediction using the Sign2Pose Gloss prediction transformer.

6.2.5 Discussion about Isolated SLR

Deep learning techniques have emerged as prominent solutions in isolated SLR. One approach involves the use of 3DCNN to capture the spatiotemporal information of sign language gestures. These models can learn both spatial features from individual frames and temporal dynamics from the sequence of frames. Another approach combines Recurrent Neural Networks (RNN), such as LSTM or GRU, with CNN to model long-term dependencies in signs. Additionally, deep transfer learning leverages pre-trained models like VGG16, Inception, and ResNet as feature extractors.

Furthermore, various techniques have been applied to address specific challenges in isolated SLR. Data augmentation, attention mechanisms, and knowledge distillation are employed to augment the training dataset, focus on relevant parts of gestures, and transfer knowledge from larger models, respectively. Multi-modal fusion and multi-view techniques are also utilized to combine information from different sources and perspectives, further improving recognition performance.

Despite the remarkable progress made in isolated SLR, several challenges remain to be addressed.

(1) The availability of large and diverse datasets

As the dataset size increases, there is a risk of overfitting, which can lead to a decrease in recognition accuracy. For example, Eunice et al. proposed Sign2Pose, achieving an accuracy of 80.9% on the WLASL100 dataset, but the accuracy dropped to 38.65% on the WLASL2000 dataset [161].

(2) The user-dependency and user-independency of recognition systems

User-dependency refers to the need for personalized models for each individual user, which can limit the scalability and practicality of the system. On the other hand, user-independency aims to develop models that can generalize well across different users. Achieving user-independency requires addressing intra-class variations and adapting the models to different signing styles and characteristics.

(3) The lack of standard evaluation metrics and benchmarks

The lack of standard evaluation metrics and benchmarks hinders the comparison and benchmarking of different approaches. Establishing standardized evaluation protocols and benchmarks would facilitate fair comparisons and advancements in the field.

6.3 Continuous Sign Language Recognition

Continuous Sign Language Recognition (CSLR) refers to the recognition and understanding of sign language in continuous and dynamic sequences, where signs are not isolated but connected together to form sentences or conversations. CSLR presents unique challenges compared to Isolated SLR, including managing temporal dynamics, variations in sign durations and speeds, co-articulation effects, incorporating non-manual features, and real-time processing. Table 7 lists a summary of CSLR techniques.

Table 7: Summary of CSLR technologies

Paper	Year	Language	Modality[1]	Database (sentences/signs/signers)	Methods	Components	Performance
[163]	2018	GSL	RGB	PHOENIX	Hybrid CNN-HMM	Full	30% (WER)
				PHOENIX14			32.5% (WER)
				SIGNUM			7.4% (WER)
[78]	2018	CSL	RGB	CSL Dataset	LS-HAN (3DCNN+ Latent Space+	Full	82.7% (Acc.)
		GSL		PHOENIX14	Hierarchical Attention Network)	H	61.7% (Acc.)
[164]	2019	GSL	RGB	PHOENIX14	CNN+ stacked temporal fusion +BiLSTM+iterative optimization	Full + OF	22.86% (WER)
				SIGNUM			2.8% (WER)
[164]	2019	GSL	RGB	PHOENIX14	Feature learning: 3D-ResNet	Full	36.7% (WER)
		CSL		CSL Dataset	sequence modelling: encoder-decoder with LSTM and CTC		32.7% (WER)
[12]	2019	ISL	Leap motion sensor	157 / 35 /6	Sub_units+2DCNN+ modified LSTM	H	72.3% (Acc.)
[166]	2019	ISL	sensor	20/- /10	CapsNet	H	94% (Acc.)
[167]	2020	GSL	RGB	PHOENIX14	Video Encoder (CNN+ stacked 1D temporal	Full	24.0% (WER)
		GSL		PHOENIX14T	convolution layers + BiLSTM) +Text		24.3% (WER)
		CSL		CSL	Encoder (LSTM)+ Latent Space Alignment +Decoder		2.4% (WER) (Split I)
[168]	2020	GSL	RGB	PHOENIX 2014 T	Multi-Stream CNN-LSTM-HMMs	H	73.4% (Acc.)
[169]	2020	GSL	RGB	PHOENIX14	CNN-TCN visual encoder, sequential model	Full	21.9% (WER)
		CSL		CSL	and text encoder, with cross modality augmentation		24.5% (WER)
[170]	2021	GSL	RGB	PHOENIX 2014 T	Spatiotemporal Feature Extractor with iteratively fine-tune sequence model (BiLSTM+CTC)	Full	34.4% (WER)
[171]	2021	GSL	RGB	PHOENIX 2014 T	GRU-RST	Full	23.5% (WER)
[172]	2021	GSL	RGB	PHOENIX14	H-GAN (LSTM+3DCNN)	Full	20.7% (WER)
[173]	2021	GSL	RGB	PHOENIX14	SMKD+ CTC	Full	21.0% (WER)
				PHOENIX14T			22.4% (WER)
[174]	2021	GSL	RGB	PHOENIX14	SignBERT (BERT+ResNet)	Full	20.2% (WER)
		CSL		CSL Dataset	BERT + ResNet		23.3% (WER)
		HKSL		HKSL			12.35% (WER)
[11]	2022	CSL	RGB	CSL Dataset	CA-SignBERT (BERT+	Full	19.8% (WER)
		GSL	RGB	PHOENIX14	cross-attention+CNN + BiLSTM+CTC loss)		18.6% (WER)
		GrSL	RGB+sm	GrSL			31.15% (WER)
		HKSL	art watch data	HKSL			7.19% (WER)

(Continued)

Table 7 (continued)

Paper	Year	Language	Modality[1]	Database (sentences/signs/signers)	Methods	Components	Performance
[175]	2022	CSL	sensor	60 /-/-	DeepSLR (attention-based Encoder-Decoder model+ multi-channel CNN)	H	10.8% (WER)
[176]	2022	GSL CSL GSL	RGB	PHOENIX14 CSL Dataset PHOENIX14T	STMC (SMC+TMC+Encoder+Decoder)	H+F+B	20.7% (WER) 28.6% (WER) 21.0% (WER)
[177]	2022	GSL CSL	RGB	PHOENIX14 CSL	two-stream Resnet34 + transformer	Full+H+F	16.72% (WER) 0.87.1% (Acc.)
[178]	2022	CSL	RGB	CSL	3D-MobileNetv2+RKD	Full	2.2% (WER) (Split I)
[179]	2022	GSL CSL	RGB	PHOENIX14 CSL	Multilingual SLR framework: CNN-TCN visual feature extractor, language-independent BLSTM-CTC branches, together with a shared BLSTM initialized with language embeddings	Full	20.9% (WER) 18.1% (WER)
[180]	2023	CSL	sensor	OH-Sentence(723/-/24) TH-Sentence(182/-/14)	SeeSign: Transformer + SA +CA	Full	18.34% (WER) 22.08% (WER)
[181]	2023	ISL	sensor	40 /-/-	CNN+BiLSTM+CTC+ transfer learning	Full	15.14 1.59
[182]	2023	CSL	RGB	CSL	TrCLR (Transformer)	Full	96.6% (Acc.)
[183]	2023	CSL	RGB	60//21000/-	STFE-Net (Bi-GRU+Transformer)	Full	–
[184]	2023	CSL	RGB skeleton data	CSL	spatial temporal graph attention network+BLSTM	Full	1.59% (WER)
[185]	2023	–	RGB	PHOENIX14 PHOENIX14	self-supervised pre-training + Downstream Fine-Tuning + multi-level masked modeling strategies	H+HP	20.0% (WER) 19.9% (WER)

Note: [1]Some databases contain multiple data modalities, such as RGB and Depth. However, not all of them are used in the algorithms. The table below only shows the modalities used in the algorithms.

6.3.1 CNN-RNN Hybrid Approaches for CLSR

The CNN-RNN hybrid approach is a powerful framework in the field of deep learning that combines the strengths of CNN and RNN. It is commonly used for tasks that involve both spatial and sequential data, such as dynamic sign video analysis.

Koller et al. [163] introduced the end-to-end embedding of a CNN into an HMM while interpreting the outputs of the CNN in a Bayesian framework. Cui et al. [164] presented a developed framework for CSLR using deep neural networks. The architecture incorporated deep CNN with stacked temporal fusion layers as the feature extraction module, and BiLSTM as the sequence-learning module. Additionally, the paper contributed to the field by exploring the multimodal fusion of RGB images and optical flow in sign language. The method was evaluated on two challenging SL recognition benchmarks: PHOENIX and SIGNUM, where it outperformed the state of the art by achieving a relative improvement of more than 15% on both databases. Mittal et al. [12] proposed a modified LSTM model for CSLR. In this model, CNN was used to extract the spatial features, A modified LSTM classifier for the recognition of continuous signed sentences using sign sub-units.

6.3.2 RNN Encoder-Decoder Approaches

The Encoder-Decoder approach is a deep learning model used for sequence-to-sequence tasks. It consists of two main components: an encoder and a decoder. In the Encoder-Decoder approach, the encoder and decoder are typically implemented using RNNs such as LSTM or GRUs. the RNN Encoder-Decoder approach is well-suited for tasks involving sequential data, where capturing dependencies and context is crucial for generating accurate and coherent output sequences.

Papastrati et al. [167] introduced a novel unified deep learning framework for vision-based CSLR. The proposed approach consisted of two encoders. a video encoder was proposed that consists of a CNN, stacked 1D temporal convolution layers (TCL), and a BiLSTM. a text encoder is implemented using a unidirectional LSTM. The outputs of both encoders are projected into a joint decoder. The proposed method on the PHOENIX14T dataset achieved a WER of 24.1% on the Dev set and 24.3% WER on the Test set. Wang et al. [175] developed a novel real-time end-to-end SLR system, named DeepSLR. The system utilized an attention-based encoder-decoder model with a multi-channel CNN. The effectiveness of DeepSLR was evaluated extensively through implementation on a smartphone and subsequent evaluations. Zhou et al. [176] proposed a spatial-temporal multi-cue (STMC) network for CSLR. The STMC network comprised a spatial multi-cue (SMC) module and a temporal multi-cue (TMC) module. They are processed by the BiLSTM encoder, the CTC decoder, and the SA-LSTM decoder for sequence learning and inference. The proposed approach achieved 20.7% WER on the test set, a new state-of-the-art result on PHOENIX14.

6.3.3 Transformer (BERT) Based Approaches for CSLR

The transformer is a neural network architecture that utilizes the encoder-decoder component structure. In the encoder-decoder framework, the transformer replaces traditional RNNs with self-attention mechanisms. This innovative approach has demonstrated impressive performance in a range of natural language processing tasks. The remarkable success of transformers in NLP has captured the interest of researchers working on SLR. Chen et al. [177] proposed a two-stream lightweight multimodal fusion sign transformer network. This approach combined the contextual capabilities of the transformer network with meaningful multimodal representations. By leveraging both visual and linguistic information, the proposed model aimed to improve the accuracy and robustness of SLR. Preliminary results of the proposed model on the PHOENIX14T dataset have

shown promising performance, with a WER of 16.72%. Zhang et al. [180] proposed SeeSign, a multimodal fusion transformer framework for SLR. SeeSign incorporated two attention mechanisms, namely statistical attention and contrastive attention, to thoroughly investigate the intra-modal and inter-modal correlations present in surface Electromyography (sEMG) and inertial measurement unit (IMU) signals, and effectively fuse the two modalities. The experimental results showed that SeeSign achieved a WER of 18.34% and 22.08% on the OH-Sentence and TH-Sentence datasets, respectively. Jiang et al. [182] presented TrCLR, a novel Transformer-based model for CSLR. To extract features, they employed the CLIP4Clip video retrieval method, while the overall model architecture adopts an end-to-end Transformer structure. The CSL dataset, consisting of sign language data, is utilized for this experiment. The experimental results demonstrated that TrCLR achieved an accuracy of 96.3%. Hu et al. [183] presented a spatial-temporal feature extraction network (STFE-Net) for continuous sign language translation (CSLT). The spatial feature extraction network (SFE-Net) selected 53 key points related to sign language from the 133 key points in the COCO-WholeBody dataset. The temporal feature extraction network (TFE-Net) utilized a Transformer to implement temporal feature extraction, incorporating relative position encoding and position-aware self-attention optimization. The proposed model achieved BLUE-1 = 77.59, BLUE-2 = 75.62, BLUE-3 = 74.25, and BLUE-4 = 72.14 on a Chinese continuous sign language dataset collected by the researchers themselves.

BERT (Bidirectional Encoder Representations from Transformers) is based on the Transformer architecture and is pre-trained on a large *corpus* of unlabeled text data [117]. This pre-training allows BERT to be fine-tuned for various NLP tasks, achieving remarkable performance across multiple domains. Zhou et al. [174] developed a deep learning framework called SignBERT. SignBERT combined the BERT with the ResNet to effectively model underlying sign languages and extract spatial features for CSLR. In another study, Zhou et al. [11] developed A BERT-based deep learning framework named CASignBERT for CSLR. the proposed CA-SignBERT framework consisted of the cross-attention mechanism and the weight control module. Experimental results demonstrated that the CA-SignBERT framework attained the lowest WER in both the validation set (18.3%) and test set (18.6%) of the PHOENIX14.

6.3.4 Other Approaches for CSLR

Some researchers used capsule networks (CapsNet), Generative Adversarial Networks (GANs), and other models for CSLR. In [166], a novel one-dimensional deep CapsNet architecture was proposed for continuous Indian SLR using signals collected from a custom-designed wearable IMU system. The performance of the proposed CapsNet architecture was evaluated by modifying the dynamic routing between capsule layers. The results showed that the proposed CapsNet achieved improved accuracy rates of 94% for 3 routings and 92.50% for 5 routings, outperforming the CNN which achieved an accuracy of 87.99%. Elakkiya et al. [172] focused on recognizing sign language gestures from continuous video sequences by characterizing manual and non-manual gestures. A novel approach called hyperparameter-based optimized GANs was introduced, which operated in three phases. In Phase I, stacked variational auto-encoders (SVAE) and Principal Component Analysis (PCA) were employed to obtain pre-tuned data with reduced feature dimensions. In Phase II, H-GANs utilized a Deep LSTM as a generator and LSTM with 3DCNN as a discriminator. The generator generated random sequences with noise based on real frames, while the discriminator detected and classified the real frames of sign gestures. In Phase III, Deep Reinforcement Learning (DRL) was employed for hyper-parameter optimization and regularization. Proximal Policy Optimization (PPO) optimized the hyper-parameters based on reward points, and Bayesian Optimization (BO) regularized the hyperparameters. Han et al. [178] built 3D CNNs such as 3D-MobileNets, 3D-ShuffleNets, and

X3Ds to create compact and fast spatiotemporal models for continuous sign language tasks. In order to enhance their performance, they also implemented a random knowledge distillation strategy (RKD).

6.3.5 Discussion about CSLR

CSLR aims to recognize and understand sign language sentences or continuous streams of gestures. This task is more challenging than isolated SLR as it involves capturing the temporal dynamics and context of the signs.

In recent years, CSLR has made significant progress, thanks to advancements in deep learning and the availability of large-scale sign language datasets. The use of 3DCNN, RNN, and their variants, and transformer models, has shown promising results in capturing the temporal dependencies and context in sign language sentences.

CSLR is a more complex and challenging task compared to isolated SLR. In CSLR, two main methods that have shown promising results are the RNN encoder-decoder and Transformer models.

The RNN encoder-decoder architecture is widely used in CSLR. The encoder processes the input sign language sequence, capturing the temporal dynamics and extracting meaningful features. Recurrent neural networks (RNN) such as LSTM or GRU are commonly used as the encoder, as they can model long-term dependencies in sign language sentences. The decoder, also an RNN, generates the output sequence, which could be a sequence of words or signs. The encoder-decoder framework allows for end-to-end training, where the model learns to encode the input sequence and generate the corresponding output sequence simultaneously.

Another notable method in CSLR is the Transformer model. Originally introduced for natural language processing tasks, the Transformer has also been adapted for SLR. The Transformer model relies on self-attention mechanisms to capture the relationships between different parts of the input sequence. It can effectively model long-range dependencies and has been shown to be highly parallelizable, making it suitable for real-time CSLR. The Transformer model has achieved competitive results in CSLR tasks and has shown potential for capturing the contextual information and syntactic structure of sign language sentences.

Despite the progress made, CSLR still faces great challenges. One major challenge is the lack of large-scale and diverse datasets for training and evaluation. Collecting and annotating continuous sign language datasets is time-consuming and requires expertise. Additionally, the variability and complexity of CSLR make it more difficult to capture and model the continuous nature of sign language. The presence of co-articulation, where signs influence each other, further complicates recognition.

Another challenge in CSLR is the need for real-time and online recognition. Unlike isolated SLR, where gestures are segmented and recognized individually, CSLR requires continuous processing and recognition of sign language sentences as they are being performed. Achieving real-time performance while maintaining high accuracy is a significant challenge that requires efficient algorithms and optimized models.

Additionally, CSLR often involves addressing the semantic and syntactic structure of sign language sentences. Sign languages have their own grammar and syntax, which need to be considered for accurate recognition. Capturing the contextual information and understanding the meaning of the signs in the context of the sentence pose additional challenges.

Furthermore, user-independency and generalization across different sign languages and users are crucial for CSLR systems. Developing models that can adapt to different signing styles, regional variations, and individual preferences is a complex task that requires extensive training data and robust algorithms.

7 Conclusions and Future Directions

In this survey paper, we have explored the application of deep learning techniques in SLR, involving fingerspelling, isolated signs, and continuous signs. We have discussed various aspects of SLR, including sign data acquisition technologies, sign language datasets, evaluation methods, and advancements in deep learning-based SLR.

CNN has been widely used in SLR, particularly for its ability to capture spatial features from individual frames. RNN, such as LSTM and GRU, has also been employed to model the temporal dynamics of sign language gestures. Both CNN and RNN have shown promising results in fingerspelling and isolated SLR. However, the continuous nature of sign language poses additional challenges, which have led to the exploration of advanced neural networks. One notable method is the Transformer model, originally introduced for natural language processing tasks but adapted for SLR. The Transformer model's self-attention mechanisms enable it to capture relationships between different parts of the input sequence, effectively modeling long-range dependencies. It has shown potential for real-time CSLR, with competitive results and the ability to capture contextual information and syntactic structure.

Despite the significant advancements in SLR using deep learning approaches, there are still several great challenges that need to be addressed in this field. Through research, we believe that there are still the following challenges in the field of SLR, as shown in Table 8.

Table 8: The main limitations in SLR

Number	Limitations	Strategies for improvement
1	Limited availability of large-scale and diverse datasets	Dataset Expansion and Diversity
2	Variability in signing styles and regional differences	Developing user-adaptive CSLR models
3	Co-articulation and spatio-temporal dependencies	Co-articulation and Temporal Modeling
4	Limited contextual information	Contextual Information Integration
5	Real-time processing and latency	Real-Time Processing Optimization
6	Generalization to new users and sign languages	Transfer learning and multilingual approaches
7	The contradiction between model accuracy and computational power	Model optimization and lightweight network architectures

(1) Limited availability of large-scale and diverse datasets

Obtaining comprehensive datasets for SLR is challenging. Scarce annotated datasets accurately representing sign language's variability and complexity restrict the training and evaluation of robust models. To overcome this, efforts should be made to collect larger-scale and more diverse datasets through collaboration among researchers, sign language communities, and organizations. Incorporating regional variations and different signing styles in the dataset can improve generalization.

Additionally, existing datasets are recorded in controlled environments, which may not accurately represent real-world conditions. Here are some suggestions for collecting sign language in more realistic contexts:

a) Record sign language conversations between two or more people in natural settings, such as a coffee shop or a park. This will help capture the nuances of sign language communication that are often missed in controlled environments;
b) Collaborate with sign language communities to collect data that is representative of their language and culture. This will ensure that the data collected is relevant and accurate.
c) Collect sign language data from social media platforms, such as YouTube or Instagram. This will allow for the collection of data from a wide range of sign language users and contexts.
d) Use crowdsourcing to collect sign language data from a large number of users. This will allow for the collection of a diverse range of data from different sign language users and contexts.

(2) Variability in signing styles and regional differences

Sign language exhibits significant variability in signing styles and regional variations, making it difficult to develop user-independent recognition systems. To enhance recognition accuracy, user-adaptive CSLR models should be developed. Personalized adaptation techniques such as user-specific fine-tuning or transfer learning can allow systems to adapt to individual signing styles. Incorporating user feedback and iterative model updates can further improve user adaptation.

(3) Co-articulation and temporal dependencies

Co-articulation, where signs influence each other, poses challenges in accurately segmenting and recognizing individual signs. To address the issue of co-articulation, it is essential to consider different modalities for SLR, including depth, thermal, Optical Flow (OF), and Scene Flow (SF), in addition to RGB image or video inputs. Multimodal fusion of these modalities can improve the accuracy and robustness of SLR systems. Furthermore, feature fusion that combines manual and non-manual features can enhance recognition accuracy. Many existing systems tend to focus solely on hand movements and neglect non-manual signs, which can lead to inaccurate recognition. To capture the continuous nature of sign language, it is crucial to model temporal dependencies effectively. Dynamic modeling approaches like hidden Markov models (HMMs) or recurrent neural networks (RNNs) can be explored to improve segmentation and recognition accuracy. These models can effectively capture the co-articulation and temporal dependencies present in sign language. da Silva et al. [186] developed a multiple-stream architecture that combines convolutional and recurrent neural networks, dealing with sign languages' visual phonemes in individual and specialized ways. The first stream uses the OF as input for capturing information about the "movement" of the sign; the second stream extracts kinematic and postural features, including "handshapes" and "facial expressions"; the third stream processes the raw RGB images to address additional attributes about the sign not captured in the previous streams.

One of the major challenges in SLR is capturing and understanding the complex spatio-temporal nature of sign language. On one hand, sign language involves dynamic movements that occur over time, and recognizing and interpreting these temporal dependencies is essential for understanding the meaning of signs. On the other hand, SLR also relies on spatial information, particularly the precise hand shape, hand orientation, and hand trajectory. To address these challenges, researchers in the field are actively exploring advanced deep learning techniques, to effectively capture and model the spatio-temporal dependencies in sign language data.

(4) Limited contextual information

Capturing and utilizing contextual information in CSLR systems remains a challenge. Understanding the meaning of signs in the context of a sentence is crucial for accurate recognition. Incorporating linguistic knowledge and language modeling techniques can help interpret signs in the context of the sentence, improving accuracy and reducing ambiguity.

(5) Real-time processing and latency

Achieving real-time and low-latency CSLR systems while maintaining high accuracy poses computational challenges. Developing efficient algorithms and optimizing computational resources can enable real-time processing. Techniques like parallel processing, model compression, and hardware acceleration should be explored to minimize latency and ensure a seamless user experience.

(6) Generalization to new users and sign languages

Generalizing CSLR models to new users and sign languages is complex. Adapting models to different users' signing styles and accommodating new sign languages require additional training data and adaptation techniques. Transfer learning can be employed to generalize CSLR models across different sign languages, reducing the need for extensive language-specific training data. Exploring multilingual CSLR models that can recognize multiple sign languages simultaneously can also improve generalization.

By addressing these limitations, the field of SLR can make significant progress in addressing the limitations and advancing the accuracy, efficiency, and generalization capabilities of SLR systems.

(7) The contradiction between model accuracy and computational power

As models become more complex and accurate, they often require a significant amount of computational power to train and deploy. This can limit their practicality and scalability in real-world applications. To address this contradiction, several approaches can be considered:

a) Explore techniques to optimize and streamline the model architecture to reduce computational requirements without sacrificing accuracy. This can include techniques like model compression, pruning, or quantization, which aim to reduce the model size and computational complexity while maintaining performance.
b) Develop lightweight network architectures specifically designed for SLR. These architectures aim to reduce the number of parameters and operations required for inference while maintaining a reasonable level of accuracy, such as [187–190].

Acknowledgement: Thanks to three anonymous reviewers and the editors of this journal for providing valuable suggestions for the paper.

Funding Statement: This work was supported from the National Philosophy and Social Sciences Foundation (Grant No. 20BTQ065).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yanqiong Zhang, Xianwei Jiang; data collection: Yanqiong Zhang; analysis and interpretation of results: Yanqiong Zhang, Xianwei Jiang; draft manuscript preparation: Yanqiong Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All the reviewed research literature and used data in this manuscript includes scholarly articles, conference proceedings, books, and reports that are publicly available. The references and citations can be found in the reference list of this manuscript and are accessible through online databases, academic libraries, or by contacting the publishers directly.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Stokoe, W. C. (1960). *Sign language structure*. Buffalo: University of Buffalo Press.
2. Batnasan, G., Gochoo, M., Otgonbold, M. E., Alnajjar, F., Shih, T. K. (2022). ArSL21L: Arabic sign language letter dataset benchmarking and an educational avatar for metaverse applications. *2022 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1814–1821. Tunis, Tunisia. https://doi.org/10.1109/EDUCON52537.2022.9766497
3. Marzouk, R., Alrowais, F., Al-Wesabi, F. N., Hilal, A. M. (2022). Atom search optimization with deep learning enabled arabic sign language recognition for speaking and hearing disability persons. *Healthcare, 10(9),* 1606. https://doi.org/10.3390/healthcare10091606
4. Amrani, N. E. A., Abra, O. E. K., Youssfi, M., Bouattane, O. (2019). A new interpretation technique of traffic signs, based on deep learning and semantic web. *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, pp. 1–6. Maui, HI, USA. https://doi.org/10.1109/ICDS47004.2019.8942319
5. Zhu, Y., Liao, M., Yang, M., Liu, W. (2018). Cascaded segmentation-detection networks for text-based traffic sign detection. *IEEE Transactions on Intelligent Transportation Systems, 19(1),* 209–219. https://doi.org/10.1109/TITS.2017.2768827
6. Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Ghadakchi, H. F. et al. (2022). Sensing and detection of traffic signs using CNNs: An assessment on their performance. *Sensors, 22(22),* 8830. https://doi.org/10.3390/s22228830
7. Manoharan, Y., Saxena, S., D., R. (2022). A vision-based smart human computer interaction system for hand-gestures recognition. *2022 1st International Conference on Computational Science and Technology (ICCST)*, pp. 321–324. Sharjah, United Arab Emirates. https://doi.org/10.1109/ICCST55948.2022.10040464
8. Hmida, I., Romdhane, N. B. (2022). Arabic sign language recognition algorithm based on deep learning for smart cities. *The 3rd International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2022)*, pp. 119–127. Sharjah, United Arab Emirates. https://doi.org/10.1049/icp.2022.2426
9. Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience, 2018,* 7068349. https://doi.org/10.1155/2018/7068349
10. Liang, Z. J., Liao, S. B., Hu, B. Z. (2018). 3D convolutional neural networks for dynamic sign language recognition. *Computer Journal, 61(11),* 1724–1736. https://doi.org/10.1093/comjnl/bxy049

11. Zhou, Z., Tam, V. W. L., Lam, E. Y. (2022). A cross-attention BERT-based framework for continuous sign language recognition. *IEEE Signal Processing Letters, 29,* 1818–1822. https://doi.org/10.1109/LSP.2022.3199665
12. Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., Chaudhuri, B. B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal, 19(16),* 7056–7063. https://doi.org/10.1109/JSEN.2019.2909837
13. Sharma, S., Kumar, K. (2021). ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimedia Tools and Applications, 80(17),* 26319–26331. https://doi.org/10.1007/s11042-021-10768-5
14. Luqman, H., El-Alfy, E. S. M. (2021). Towards hybrid multimodal manual and non-manual Arabic sign language recognition: mArSL database and pilot study. *Electronics, 10(14),* 1739. https://doi.org/10.3390/electronics10141739
15. Xue, C., Yu, M., Yan, G., Qin, M., Liu, Y. et al. (2022). A multi-modal fusion framework for continuous sign language recognition based on multi-layer self-attention mechanism. *Journal of Intelligent & Fuzzy Systems, 43(4),* 4303–4316. https://doi.org/10.3233/JIFS-211697
16. Kudrinko, K., Flavin, E., Zhu, X., Li, Q. (2021). Wearable sensor-based sign language recognition: A comprehensive review. *IEEE Reviews in Biomedical Engineering, 14,* 82–97. https://doi.org/10.1109/RBME.2020.3019769
17. Ahmed, M. A., Zaidan, B. B., Zaidan, A. A., Salih, M. M., Lakulu, M. M. B. (2018). A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors, 18(7),* 2208. https://doi.org/10.3390/s18072208
18. Lu, C., Amino, S., Jing, L. (2023). Data glove with bending sensor and inertial sensor based on weighted DTW fusion for sign language recognition. *Electronics, 12(3),* 613. https://doi.org/10.3390/electronics12030613
19. Alzubaidi, M. A., Otoom, M., Abu Rwaq, A. M. (2023). A novel assistive glove to convert arabic sign language into speech. *ACM Transactions on Asian and Low-Resource Language Information Processing, 22(2),* 1–16. https://doi.org/10.1145/3545113
20. DelPreto, J., Hughes, J., D'Aria, M., de Fazio, M., Rus, D. (2022). A wearable smart glove and its application of pose and gesture detection to sign language classification. *IEEE Robotics and Automation Letters, 7(4),* 10589–10596. https://doi.org/10.1109/LRA.2022.3191232
21. Oz, C., Leu, M. C. (2011). American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence, 24(7),* 1204–1213. https://doi.org/10.1016/j.engappai.2011.06.015
22. Dias, T. S., Alves Mendes Junior, J. J., Pichorim, S. F. (2022). An instrumented glove for recognition of Brazilian Sign Language Alphabet. *IEEE Sensors Journal, 22(3),* 2518–2529. https://doi.org/10.1109/JSEN.2021.3136790
23. Wen, F., Zhang, Z., He, T., Lee, C. (2021). AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove. *Nature Communications, 12(1),* Article 1. https://doi.org/10.1038/s41467-021-25637-w
24. Lee, C. K. M., Ng, K. K. H., Chen, C. H., Lau, H. C. W., Chung, S. Y. et al. (2021). American sign language recognition and training method with recurrent neural network. *Expert Systems with Applications, 167,* 114403. https://doi.org/10.1016/j.eswa.2020.114403
25. Abdullahi, S. B., Chamnongthai, K. (2022). American sign language words recognition of skeletal videos using processed video driven multi-stacked deep LSTM. *Sensors, 22(4),* 1406. https://doi.org/10.3390/s22041406
26. Abdullahi, S. B., Chamnongthai, K. (2022). American sign language words recognition using spatio-temporal prosodic and angle features: A sequential learning approach. *IEEE Access, 10,* 15911–15923. https://doi.org/10.1109/ACCESS.2022.3148132

27. Li, J., Zhong, J., Wang, N. (2023). A multimodal human-robot sign language interaction framework applied in social robots. *Frontiers in Neuroscience, 17,* 1168888. https://doi.org/10.3389/fnins.2023.1168888
28. Pacifici, I., Sernani, P., Falcionelli, N., Tomassini, S., Dragoni, A. F. (2020). A surface electromyography and inertial measurement unit dataset for the Italian Sign Language alphabet. *Data in Brief, 33,* 106455 https://doi.org/10.1016/j.dib.2020.106455
29. Mendes Junior, J. J. A., Freitas, M. L. B., Campos, D. P., Farinelli, F. A., Stevan, S. L. et al. (2020). Analysis of influence of segmentation, features, and classification in sEMG processing: A case study of recognition of Brazilian sign language alphabet. *Sensors, 20(16),* 4359. https://doi.org/10.3390/s20164359
30. Gu, Y., Zheng, C., Todoh, M., Zha, F. (2022). American sign language translation using wearable inertial and electromyography sensors for tracking hand movements and facial expressions. *Frontiers in Neuroscience, 16,* 962141. https://doi.org/10.3389/fnins.2022.962141
31. Tateno, S., Liu, H., Ou, J. (2020). Development of Sign Language Motion recognition system for hearing-impaired people using electromyography signal. *Sensors, 20(20),* 5807. https://doi.org/10.3390/s20205807
32. Khomami, S. A., Shamekhi, S. (2021). Persian sign language recognition using IMU and surface EMG sensors. *Measurement, 168,* https://doi.org/10.1016/j.measurement.2020.108471
33. Zhang, N., Zhang, J., Ying, Y., Luo, C., Li, J. (2022). Wi-Phrase: Deep residual-multihead model for WiFi sign language phrase recognition. *IEEE Internet of Things Journal, 9(18),* 18015–18027. https://doi.org/10.1109/JIOT.2022.3164243
34. Thariq Ahmed, H. F., Ahmad, H., Phang, S. K., Harkat, H., Narasingamurthi, K. (2021). Wi-Fi CSI based human sign language recognition using LSTM network. *2021 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 51–57. Bandung, Indonesia. https://doi.org/10.1109/IAICT52856.2021.9532548
35. Zhang, L., Zhang, Y., Zheng, X. (2020). WiSign: Ubiquitous American sign language recognition using commercial Wi-Fi devices. *ACM Transactions on Intelligent Systems and Technology, 11(3),* 1–24. https://doi.org/10.1145/3377553
36. Chen, H., Feng, D., Hao, Z., Dang, X., Niu, J. et al. (2022). Air-CSL: Chinese sign language recognition based on the commercial WiFi devices. *Wireless Communications and Mobile Computing, 2022,* 5885475. https://doi.org/10.1155/2022/5885475
37. Gurbuz, S. Z., Rahman, M. M., Kurtoglu, E., Malaia, E., Gurbuz, A. C. et al. (2022). Multi-frequency RF sensor fusion for word-level fluent ASL recognition. *IEEE Sensors Journal, 22(12),* 11373–11381. https://doi.org/10.1109/JSEN.2021.3078339
38. Hameed, H., Usman, M., Khan, M. Z., Hussain, A., Abbas., H. et al. (2022). Privacy-preserving british sign language recognition using deep learning. *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4316–4319. Glasgow, Scotland, UK. https://doi.org/10.1109/EMBC48229.2022.9871491
39. Kulhandjian, H., Sharma, P., Kulhandjian, M., D'Amours, C. (2019). Sign language gesture recognition using doppler radar and deep learning. *2019 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6. Waikoloa, HI, USA. https://doi.org/10.1109/GCWkshps45667.2019.9024607
40. McCleary, J., García, L. P., Ilioudis, C., Clemente, C. (2021). Sign language recognition using micro-doppler and explainable deep learning. *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6. Atlanta, GA, USA. https://doi.org/10.1109/RadarConf2147009.2021.9455257
41. Rahman, M. M., Mdrafi, R., Gurbuz, A. C., Malaia, E., Crawford., C. et al. (2021). Word-level sign language recognition using linguistic adaptation of 77 GHz FMCW radar data. *2021 IEEE Radar Conference (RadarConf21)*, pp. 1–6. https://doi.org/10.1109/RadarConf2147009.2021.9455190
42. Cerna, L. R., Cardenas, E. E., Miranda, D. G., Menotti, D., Camara-Chavez, G. (2021). A multimodal LIBRAS-UFOP Brazilian sign language dataset of minimal pairs using a microsoft Kinect sensor. *Expert Systems With Applications, 167,* 114179. https://doi.org/10.1016/j.eswa.2020.114179

43. Lee, G. C., Yeh, F. H., Hsiao, Y. H. (2016). Kinect-based Taiwanese sign-language recognition system. *Multimedia Tools And Applications, 75(1),* 261–279. https://doi.org/10.1007/s11042-014-2290-x
44. Sun, C., Zhang, T., Xu, C. (2015). Latent support vector machine modeling for sign language recognition with kinect. *ACM Transactions on Intelligent Systems and Technology, 6(2),* 1–20. https://doi.org/10.1145/2629481
45. Ansari, Z. A., Harit, G. (2016). Nearest neighbour classification of Indian sign language gestures using kinect camera. *Sadhana-Academy Proceedings in Engineering Sciences, 41(2),* 161–182. https://doi.org/10.1007/s12046-015-0405-3
46. Raghuveera, T., Deepthi, R., Mangalashri, R., Akshaya, R. (2020). A depth-based Indian sign language recognition using Microsoft Kinect. *Sādhanā, 45(1),* 34. https://doi.org/10.1007/s12046-019-1250-6
47. Gangrade, J., Bharti, J., Mulye, A. (2022). Recognition of Indian sign language using ORB with bag of visual words by Kinect sensor. *IETE Journal of Research, 68(4),* 2953–2967. https://doi.org/10.1080/03772063.2020.1739569
48. Yang, H. D. (2015). Sign language recognition with the kinect sensor based on conditional random fields. *Sensors, 15(1),* 135–147. https://doi.org/10.3390/s150100135
49. Kraljević, L., Russo, M., Pauković, M., Šarić, M. (2020). A dynamic gesture recognition interface for smart home control based on croatian sign language. *Applied Sciences, 10(7)*. https://doi.org/10.3390/app10072300
50. Pugeault, N., Bowden, R. (2011). Spelling it out: Real-time ASL fingerspelling recognition. *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops*, Barcelona, Spain.
51. Wikipedia (2023). ASL fingerspelling. https://en.wikipedia.org/wiki/American_manual_alphabet (accessed on 14/10/2023)
52. Camgoz, N. C., Hadfield, S., Koller, O., Hermann, N. RWTH-PHOENIX-Weather (2014). T: Parallel corpus of sign language video, gloss and translation. https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/ (accessed on 20/10/2023)
53. Paudyal, P. (2018). "American sign language (ASL) Fingerspelling dataset for Myo Sensor", *Mendeley Data*. https://doi.org/10.17632/dbymbhhpk9.1
54. Shi, B., Del Rio, A. M., Keane, J., Michaux, J. et al. (2018). American sign language fingerspelling recognition in the wild. *2018 IEEE Workshop on Spoken Language Technology (SLT 2018)*, pp. 145–152. Athens, Greece.
55. Bowen, S., Aurora, M. D. R., Jonathan, K., Jonathan, M., Diane, B. et al. (2018). Chicago Fingerspelling in the Wild Data Sets (ChicagoFSWild, ChicagoFSWild+). https://home.ttic.edu/~klivescu/ChicagoFSWild.htm (accessed on 23/10/2023)
56. Gao, Y., Zhang, Y., Jiang, X. (2022). An optimized convolutional neural network with combination blocks for chinese sign language identification. *Computer Modeling in Engineering & Sciences, 132(1),* 95–117. https://doi.org/10.32604/cmes.2022.019970
57. Latif, G., Mohammad, N., Alghazo, J., AlKhalaf, R., AlKhalaf, R. (2019). ArASL: Arabic alphabets sign language dataset. *Data in Brief, 23,* 103777. https://doi.org/10.1016/j.dib.2019.103777
58. Munkhjargal, G. (2018). ArSL. https://www.kaggle.com/datasets/alaatamimi/arsl2018 (accessed on 21/10/2023)
59. Joze, H. R. V., Koller, O. (2018). MS-ASL: A large-scale data set and benchmark for understanding American sign language. https://doi.org/10.48550/arXiv.1812.01053
60. Oscar, K. (2019). Papers with code—MS-ASL dataset. https://paperswithcode.com/dataset/ms-asl (accessed on 21/10/2023)
61. Huang, J., Zhou, W., Li, H., Li, W. (2019). Attention-based 3D-CNNs for large-vocabulary sign language recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 29(9),* 2822–2832. https://doi.org/10.1109/TCSVT.2018.2870740

62. Huang, J., Zhou, W., Li, H., Li, W. (2019). Papers with code—CSL dataset. https://paperswithcode.com/dataset/csl (accessed on 21/10/2023)
63. Sridhar, A., Ganesan, R. G., Kumar, P., Khapra, M. (2020). INCLUDE: A large scale dataset for Indian sign language recognition. *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1366–1375. Seattle, WA, USA. https://doi.org/10.1145/3394171.3413528
64. Sridhar, A., Ganesan, R. G., Kumar, P., Khapra, M. (2020). INCLUDE. https://zenodo.org/records/4010759 (accessed on 14/10/2023)
65. Li, D., Opazo, C. R., Yu, X., Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1448–1458. Snowmass, CO, USA. https://doi.org/10.1109/WACV45572.2020.9093512
66. Dongxu (2023). WLASL: A large-scale dataset for Word-Level American sign language (WACV 20' Best Paper Honourable Mention). https://github.com/dxli94/WLASL (accessed on 21/10/2023)
67. Sincan, O. M., Keles, H. Y. (2020). AUTSL: A large scale multi-modal turkish sign language dataset and baseline methods. *IEEE Access, 8,* 181340–181355. https://doi.org/10.1109/ACCESS.2020.3028072
68. Sincan, O. M., Keles, H. Y. (2020). AUTSL Dataset. http://cvml.ankara.edu.tr/datasets/ (accessed on 21/10/2023)
69. Rezende, T. M., Moreira Almeida, S. G., Guimaraes, F. G. (2021). Development and validation of a Brazilian sign language database for human gesture recognition. *Neural Computing & Applications, 33(16),* 10449–10467. https://doi.org/10.1007/s00521-021-05802-4
70. Rezende, T. M., Moreira Almeida, S. G., Guimaraes, F. G. (2021). Libras. https://dataportal.asia/dataset/212582112_libras-movement (accessed on 21/10/2023)
71. Sidig, A. A. I., Luqman, H., Mahmoud, S., Mohandes, M. (2021). KArSL: Arabic sign language database. *ACM Transactions on Asian and Low-Resource Language Information Processing, 20(1),* 1–19. https://doi.org/10.1145/3423420
72. Sidig, A. A. I., Luqman, H., Mahmoud, S., Mohandes, M. (2021). KArSL. https://github.com/Hamzah-Luqman/KArSL (accessed on 14/10/2023)
73. Islam, M. D. M., Uddin, M. D. R., Ferdous, M. J., Akter, S., Nasim Akhtar, M. D. (2022). BdSLW-11: Dataset of Bangladeshi sign language words for recognizing 11 daily useful BdSL words. *Data in Brief, 45,* 108747. https://doi.org/10.1016/j.dib.2022.108747
74. Islam, M. D. M., Uddin, M. D. R., Ferdous, M. J., Akter, S., Nasim Akhtar, M. D. (2022). BdSLW-11: A bangladeshi sign language words dataset for recognizing 11 daily useful BdSL words—Mendeley data. https://data.mendeley.com/datasets/523d6dxz4n/4 (accessed on 21/10/2023)
75. Forster, J., Schmidt, C., Hoyoux, T., Koller, O., Zelle, U. et al. (2012). RWTH-PHOENIX-weather: A large vocabulary sign language recognition and translation corpus. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*. Istanbul, Turkey.
76. Forster, J., Schmidt, C., Roller, O., Bellgardt, M., Ney, H. (2014). Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-weather. *9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.
77. Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., Bowden, R. (2018). Neural sign language translation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7784–7793. Salt Lake City, UT, USA. https://doi.org/10.1109/CVPR.2018.00812
78. Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W. (2018). Video-based sign language recognition without temporal segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, New Orleans, Louisiana, USA. https://doi.org/10.1609/aaai.v32i1.11903
79. Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W. (2018). Csl_daily. https://ustc-slr.github.io/datasets/2021_csl_daily/ (accessed on 21/10/2023)

80. Agris, U. V., Knorr, M., Kraiss, K. F. (2009). The significance of facial features for automatic sign language recognition. *IEEE International Conference on Automatic Face & Gesture Recognition*. Amsterdam, Netherlands.
81. Agris, U. V., Knorr, M., Kraiss, K. F. (2009). SIGNUM database–ELRA catalogue. (n.d.). https://catalogue.elra.info/en-us/repository/browse/ELRA-S0300/ (accessed on 21/10/2023)
82. Joze, H. R. V. (2018). MS-ASL: A large-scale data set and benchmark for understanding American sign language. https://doi.org/10.48550/arXiv.1812.01053
83. Ma, Y., Xu, T., Kim, K. (2022). A digital sign language recognition based on a 3D-CNN system with an attention mechanism. *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 1–4. Yeosu, Korea. https://doi.org/10.1109/ICCE-Asia57006.2022.9954810
84. Ji, S., Xu, W., Yang, M., Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1),* 221–231. https://doi.org/10.1109/TPAMI.2012.59
85. Lu, Z., Qin, S., Li, X., Li, L., Zhang, D. (2019). One-shot learning hand gesture recognition based on modified 3d convolutional neural networks. *Machine Vision and Applications, 30(7–8),* 1157–1180. https://doi.org/10.1007/s00138-019-01043-7
86. Boukdir, A., Benaddy, M., Ellahyani, A., El Meslouhi, O., Kardouchi, M. (2022). 3D gesture segmentation for word-level Arabic sign language using large-scale RGB video sequences and autoencoder convolutional networks. *Signal Image and Video Processing, 16(8),* 2055–2062. https://doi.org/10.1007/s11760-022-02167-6
87. Ren, X., Xiang, L., Nie, D., Shao, Y., Zhang, H. et al. (2018). Interleaved 3D-CNNs for joint segmentation of small-volume structures in head and neck CT images. *Medical Physics, 45(5),* 2063–2075. https://doi.org/10.1002/mp.12837
88. Ling, N. Z. (2019). Convolutional neural network (CNN) detailed explanation. https://www.cnblogs.com/LXP-Never/p/9977973.html (accessed on 03/07/2023)
89. Liu, C. (2019). RNN. https://blog.csdn.net/qq_32505207/article/details/105227028 (accessed on 30/08/2023)
90. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8),* 1735–1780.
91. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F. et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. http://arxiv.org/abs/1406.1078
92. Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. http://arxiv.org/abs/1412.3555
93. Schuster, M., Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 8,* 2673–2681.
94. Schuster, M. (2021). Bidirectional RNN. https://blog.csdn.net/csdn_xmj/article/details/118195670 (accessed on 18/07/2023)
95. Mitchell, T. (1997). *Machine learning*. McGraw Hill.
96. Pan, S. J., Tsang, I. W., Kwok, J. T., Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks, 22(2),* 199–210. https://doi.org/10.1109/TNN.2010.2091281
97. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C. et al. (2018). A survey on deep transfer learning. *27th International Conference on Artificial Neural Networks*, Rhodes, Greece.
98. Ganin, Y., Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1180–1189.
99. Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. https://doi.org/10.48550/arXiv.1409.1556

100. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. et al. (2014). Going deeper with convolutions. https://doi.org/10.48550/arXiv.1409.4842
101. Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. https://doi.org/10.48550/arXiv.1502.03167
102. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015). Rethinking the inception architecture for computer vision. https://doi.org/10.48550/arXiv.1512.00567
103. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. http://arxiv.org/abs/1602.07261
104. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. Las Vegas, NV, USA. https://doi.org/10.1109/CVPR.2016.90
105. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K. (2017). Aggregated residual transformations for deep neural networks. *Computer Vision and Pattern Recognition, 8,* 5987–5995.
106. Zagoruyko, S., Komodakis, N. (2016). Wide residual networks. https://doi.org/10.48550/arXiv.1605.07146
107. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q. (2018). Densely connected convolutional networks. https://doi.org/10.48550/arXiv.1608.06993
108. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861
109. Zhang, X., Zhou, X., Lin, M., Sun, J. (2017). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. http://arxiv.org/abs/1707.01083
110. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. et al. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. https://doi.org/10.48550/arXiv.1602.07360
111. Tan, M., Le, Q. V. (2020). EfficientNet: Rethinking model scaling for convolutional neural networks. https://doi.org/10.48550/arXiv.1905.11946
112. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. http://arxiv.org/abs/1610.02357
113. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2019). MobileNetV2: Inverted residuals and linear bottlenecks. https://doi.org/10.48550/arXiv.1801.04381
114. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B. et al. (2019). Searching for MobileNetV3. https://doi.org/10.48550/arXiv.1905.02244
115. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. (2017). Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762
116. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training. https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (accessed on 21/10/2023)
117. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/arXiv.1810.04805
118. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X. et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. https://doi.org/10.48550/arXiv.2010.11929
119. Jiang, X., Zhang, Y. D. (2019). Chinese sign language fingerspelling recognition via six-layer convolutional neural network with leaky rectified linear units for therapy and rehabilitation. *Journal of Medical Imaging and Health Informatics, 9(9),* 2031–2038. https://doi.org/10.1166/jmihi.2019.2804
120. Aly, W., Aly, S., Almotairi, S. (2019). User-independent American sign language alphabet recognition based on depth image and PCANet features. *IEEE Access, 7,* 123138–123150. https://doi.org/10.1109/ACCESS.2019.2938829
121. Jiang, X., Lu, M., Wang, S. H. (2020). An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of Chinese sign language. *Multimedia Tools and Applications, 79(21–22),* 15697–15715. https://doi.org/10.1007/s11042-019-08345-y

122. Nihal, R. A., Rahman, S., Broti, N. M., Deowan, S. A. (2021). Bangla sign alphabet recognition with zero-shot and transfer learning. *Pattern Recognition Letters, 150,* 84–93. https://doi.org/10.1016/j.patrec.2021.06.020
123. Martinez-Martin, E., Morillas-Espejo, F. (2021). Deep learning techniques for spanish sign language interpretation. *Computational Intelligence and Neuroscience, 2021*. https://doi.org/10.1155/2021/5532580
124. Aksoy, B., Salman, O. K. M., Ekrem, O. (2021). Detection of Turkish sign language using deep learning and image processing methods. *Applied Artificial Intelligence, 35(12),* 952–981. https://doi.org/10.1080/08839514.2021.1982184
125. Pariwat, T., Seresangtakul, P. (2021). Multi-stroke thai finger-spelling sign language recognition system with deep learning. *Symmetry, 13(2),* 262. https://doi.org/10.3390/sym13020262
126. Rivera-Acosta, M., Ruiz-Varela, J. M., Ortega-Cisneros, S., Rivera, J., Parra-Michel, R. et al. (2021). Spelling correction real-time american sign language alphabet translation system based on YOLO network and LSTM. *Electronics, 10(9)*. https://doi.org/10.3390/electronics10091035
127. Podder, K. K., Chowdhury, M. E. H., Tahir, A. M., Mahbub, Z. B., Khandakar, A. et al. (2022). Bangla Sign Language (BdSL) alphabets and numerals classification using a deep learning model. *Sensors, 22(2),* 574. https://doi.org/10.3390/s22020574
128. Ma, Y., Xu, T., Han, S., Kim, K. (2022). Ensemble learning of multiple deep CNNs using accuracy-based weighted voting for ASL recognition. *Applied Sciences, 12(22),* 111. https://doi.org/10.3390/app122211766
129. Zhang, Y., Xu, W., Zhang, X., Li, L. (2022). Sign annotation generation to alphabets via integrating visual data with somatosensory data from flexible strain sensor-based data glove. *Measurement, 202,* 111700. https://doi.org/10.1016/j.measurement.2022.111700
130. Gu, Y., Sherrine, S., Wei, W., Li, X., Yuan, J. et al. (2022). American sign language alphabet recognition using inertial motion capture system with deep learning. *Inventions, 7(4),* 112. https://doi.org/10.3390/inventions7040112
131. Kasapbaşi, A., Elbushra, A. E. A., Al-hardanee, O., Yilmaz, A. (2022). DeepASLR: A CNN based human computer interface for American sign language recognition for hearing-impaired individuals. *Computer Methods and Programs in Biomedicine Update, 2,* 100048. https://doi.org/10.1016/j.cmpbup.2021.100048
132. Das, S., Imtiaz, Md S., Neom, N. H., Siddique, N., Wang, H. (2023). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. *Expert Systems With Applications, 213,* 118914. https://doi.org/10.1016/j.eswa.2022.118914
133. Nareshkumar, M. D., Jaison, B. (2023). A light-weight deep learning-based architecture for sign language classification. *Intelligent Automation and Soft Computing, 35(3),* 3501–3515. https://doi.org/10.32604/iasc.2023.027848
134. Aljuhani, R., Alfaidi, A., Alshehri, B., Alwadei, H., Aldhahri, E. et al. (2023). Arabic sign language recognition using convolutional neural network and mobileNet. *Arabian Journal for Science and Engineering, 48(2),* 2147–2154. https://doi.org/10.1007/s13369-022-07144-2
135. Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., Gogoi, D. (2023). Real-time assamese sign language recognition using mediapipe and deep learning. *International Conference on Machine Learning and Data Engineering, 218,* 1384–1393. https://doi.org/10.1016/j.procs.2023.01.117
136. Kothadiya, D. R., Bhatt, C. M., Saba, T., Rehman, A., Bahaj, S. A. (2023). SIGNFORMER: DeepVision transformer for sign language recognition. *IEEE Access, 11,* 4730–4739. https://doi.org/10.1109/ACCESS.2022.3231130
137. Nandi, U., Ghorai, A., Singh, M. M., Changdar, C., Bhakta, S. et al. (2023). Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling. *Multimedia Tools and Applications, 82(7),* 9627–9648. https://doi.org/10.1007/s11042-021-11595-4
138. Zhang, L., Tian, Q., Ruan, Q., Shi, Z. (2023). A simple and effective static gesture recognition method based on attention mechanism. *Journal of Visual Communication and Image Representation, 92,* 103783. https://doi.org/10.1016/j.jvcir.2023.103783

139. Ang, M. C., Taguibao, K. R. C., Manlises, C. O. (2022). Hand gesture recognition for Filipino sign language under different backgrounds. *2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pp. 1–6. Kota Kinabalu, Malaysia. https://doi.org/10.1109/IICAIET55139.2022.9936801
140. Siddique, S., Islam, S., Neon, E. E., Sabbir, T., Naheen, I. T. et al. (2023). Deep learning-based Bangla sign language detection with an edge device. *Intelligent Systems with Applications, 18,* 200224. https://doi.org/10.1016/j.iswa.2023.200224
141. Jiang, X., Satapathy, S. C., Yang, L., Wang, S. H., Zhang, Y. D. (2020). A survey on artificial intelligence in Chinese sign language recognition. *Arabian Journal for Science and Engineering, 45(12),* 9859–9894. https://doi.org/10.1007/s13369-020-04758-2
142. Lee, B. G., Chong, T. W., Chung, W. Y. (2020). Sensor fusion of motion-based sign language interpretation with deep learning. *Sensors, 20(21),* 6256. https://doi.org/10.3390/s20216256
143. Aly, S., Aly, W. (2020). DeepArSLR: A Novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition. *IEEE Access, 8,* 83199–83212. https://doi.org/10.1109/ACCESS.2020.2990699
144. Yu, Y., Chen, X., Cao, S., Zhang, X., Chen, X. (2020). Exploration of Chinese sign language recognition using wearable sensors based on deep Belief Net. *IEEE Journal of Biomedical and Health Informatics, 24(5),* 1310–1320. https://doi.org/10.1109/JBHI.2019.2941535
145. Rastgoo, R., Kiani, K., Escalera, S. (2020). Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools and Applications, 79(31–32),* 22965–22987. https://doi.org/10.1007/s11042-020-09048-5
146. Venugopalan, A., Reghunadhan, R. (2021). Applying deep neural networks for the automatic recognition of sign language words: A communication aid to deaf agriculturists. *Expert Systems with Applications, 185,* 115601. https://doi.org/10.1016/j.eswa.2021.115601
147. Shaik, A. A., Mareedu, V. D. P., Polurie, V. V. K. (2021). Learning multiview deep features from skeletal sign language videos for recognition. *Turkish Journal of Electrical Engineering and Computer Sciences, 29(2),* 1061–1076. https://doi.org/10.3906/elk-2005-57
148. Abdul, W., Alsulaiman, M., Amin, S. U., Faisal, M., Muhammad, G. et al. (2021). Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM. *Computers and Electrical Engineering, 95,* 107395. https://doi.org/10.1016/j.compeleceng.2021.107395
149. Rastgoo, R., Kiani, K., Escalera, S. (2021). Hand pose aware multimodal isolated sign language recognition. *Multimedia Tools and Applications, 80(1),* 127–163. https://doi.org/10.1007/s11042-020-09700-0
150. Boukdir, A., Benaddy, M., Ellahyani, A., El Meslouhi, O., Kardouchi, M. (2022). Isolated video-based Arabic sign language recognition using convolutional and recursive neural networks. *Arabian Journal for Science and Engineering, 47(2),* 2187–2199. https://doi.org/10.1007/s13369-021-06167-5
151. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-Gonzalez, A. B. et al. (2022). Deepsign: Sign language detection and recognition using deep learning. *Electronics, 11(11),* 1780. https://doi.org/10.3390/electronics11111780
152. Guney, S., Erkus, M. (2022). A real-time approach to recognition of Turkish sign language by using convolutional neural networks. *Neural Computing & Applications, 34(5),* 4069–4079. https://doi.org/10.1007/s00521-021-06664-6
153. Li, H., Zhang, Y., Cao, Q. (2022). MyoTac: Real-time recognition of tactical sign language based on lightweight deep neural network. *Wireless Communications & Mobile Computing, 2022*. https://doi.org/10.1155/2022/2774430
154. Venugopalan, A., Reghunadhan, R. (2023). Applying hybrid deep neural network for the recognition of sign language words used by the deaf COVID-19 patients. *Arabian Journal for Science and Engineering, 48(2),* 1349–1362. https://doi.org/10.1007/s13369-022-06843-0

155. Balaha, M. M., El-Kady, S., Balaha, H. M., Salama, M., Emad, E. et al. (2023). A vision-based deep learning approach for independent-users Arabic sign language interpretation. *Multimedia Tools and Applications, 82(5),* 6807–6826. https://doi.org/10.1007/s11042-022-13423-9

156. Abdallah, M. S. S., Samaan, G. H. H., Wadie, A. R. R., Makhmudov, F., Cho, Y. I. (2023). Light-weight deep learning techniques with advanced processing for real-time hand gesture recognition. *Sensors, 23(1),* 2. https://doi.org/10.3390/s23010002

157. Gupta, R., Bhatnagar, A. S., Singh, G. (2023). A weighted deep ensemble for Indian sign language recognition. *IETE Journal of Research,* https://doi.org/10.1080/03772063.2023.2175057

158. Das, S., Biswas, S. K., Purkayastha, B. (2023). A deep sign language recognition system for Indian sign language. *Neural Computing & Applications, 35(2),* 1469–1481. https://doi.org/10.1007/s00521-022-07840-y

159. Ozdemir, O., Baytas, I. M., Akarun, L. (2023). Multi-cue temporal modeling for skeleton-based sign language recognition. *Frontiers in Neuroscience, 17,* 1148191. https://doi.org/10.3389/fnins.2023.1148191

160. Miah, A. S. M., Shin, J., Hasan, M. A. M., Rahim, M. A., Okuyama, Y. (2023). Rotation, translation and scale invariant Sign word recognition using deep learning. *Computer Systems Science and Engineering, 44(3),* 2521–2536. https://doi.org/10.32604/csse.2023.029336

161. Eunice, J., Andrew, J., Sei, Y., Hemanth, D. J. (2023). Sign2Pose: A pose-based approach for gloss prediction using a transformer model. *Sensors, 23(5),* 2853. https://doi.org/10.3390/s23052853

162. Rajalakshmi, E., Elakkiya, R., Prikhodko, A. L., Grif, M. G., Bakaev, M. A. et al. (2023). Static and dynamic isolated indian and Russian sign language recognition with spatial and temporal feature detection using hybrid neural network. *ACM Transactions on Asian and Low-Resource Language Information Processing, 22(1),* 26. https://doi.org/10.1145/3530989

163. Koller, O., Zargaran, S., Ney, H., Bowden, R. (2018). Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision, 126(12),* 1311–1325. https://doi.org/10.1007/s11263-018-1121-3

164. Cui, R., Liu, H., Zhang, C. (2019). A deep neural framework for continuous sign language recognition by iterative training. *IEEE Transactions on Multimedia, 21(7),* 1880–1891. https://doi.org/10.1109/TMM.2018.2889563

165. Pu, J., Zhou, W., Li, H. (2019). Iterative alignment network for continuous sign language recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*, pp. 4160–4169. https://doi.org/10.1109/CVPR.2019.00429

166. Suri, K., Gupta, R. (2019). Continuous sign language recognition from wearable IMUs using deep capsule networks and game theory. *Computers & Electrical Engineering, 78,* 493–503. https://doi.org/10.1016/j.compeleceng.2019.08.006

167. Papastratis, I., Dimitropoulos, K., Konstantinidis, D., Daras, P. (2020). Continuous sign language recognition through cross-modal alignment of video and text embeddings in a joint-latent space. *IEEE Access, 8,* 91170–91180. https://doi.org/10.1109/ACCESS.2020.2993650

168. Koller, O., Camgoz, N. C., Ney, H., Bowden, R. (2020). Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(9),* 2306–2320. https://doi.org/10.1109/TPAMI.2019.2911077

169. Pu, J., Zhou, W., Hu, H., Li, H., Assoc Comp Machinery (2020). Boosting continuous sign language recognition via cross modality augmentation. *Chinese Academy of Sciences,* 1497–1505. https://doi.org/10.1145/3394171.3413931

170. Koishybay, K., Mukushev, M., Sandygulova, A. (2021). Continuous sign language recognition with iterative spatiotemporal fine-tuning. *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10211–10218. Milan, Italy. https://doi.org/10.1109/ICPR48806.2021.9412364

171. Aloysius, N., G., M., Nedungadi, P. (2021). Incorporating relative position information in transformer-based sign language recognition and translation. *IEEE Access, 9,* 145929–145942. https://doi.org/10.1109/ACCESS.2021.3122921

172. Elakkiya, R., Vijayakumar, P., Kumar, N. (2021). An optimized generative adversarial network based continuous sign language classification. *Expert Systems with Applications, 182,* 115276. https://doi.org/10.1016/j.eswa.2021.115276
173. Hao, A., Min, Y., Chen, X. (2021). Self-mutual distillation learning for continuous sign language recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11283–11292. Montreal, QC, Canada. https://doi.org/10.1109/ICCV48922.2021.01111
174. Zhou, Z., Tam, V. W. L., Lam, E. Y. (2021). SignBERT: A BERT-based deep learning framework for continuous sign language recognition. *IEEE Access, 9,* 161669–161682. https://doi.org/10.1109/ACCESS.2021.3132668
175. Wang, Z., Zhao, T., Ma, J., Chen, H., Liu, K. et al. (2022). Hear sign language: A real-time end-to-end sign language recognition system. *IEEE Transactions on Mobile Computing, 21(7),* 2398–2410. https://doi.org/10.1109/TMC.2020.3038303
176. Zhou, H., Zhou, W., Zhou, Y., Li, H. (2022). Spatial-temporal multi-cue network for sign language recognition and translation. *IEEE Transactions on Multimedia, 24,* 768–779. https://doi.org/10.1109/TMM.2021.3059098
177. Chen, Y., Mei, X., Qin, X. (2022). Two-stream lightweight sign language transformer. *Machine Vision and Applications, 33(5),* 79. https://doi.org/10.1007/s00138-022-01330-w
178. Han, X., Lu, F., Tian, G. (2022). Efficient 3D CNNs with knowledge transfer for sign language recognition. *Multimedia Tools and Applications, 81(7),* 10071–10090. https://doi.org/10.1007/s11042-022-12051-7
179. Hu, H., Pu, J., Zhou, W., Li, H. (2022). Collaborative multilingual continuous sign language recognition: A unified framework. *IEEE Transactions on Multimedia,* 1–12. https://doi.org/10.1109/TMM.2022.3223260
180. Zhang, J., Wang, Q., Wang, Q., Zheng, Z. (2023). Multimodal fusion framework based on statistical attention and contrastive attention for sign language recognition. *IEEE Transactions on Mobile Computing,* 1–13. https://doi.org/10.1109/TMC.2023.3235935
181. Sharma, S., Gupta, R., Kumar, A. (2023). Continuous sign language recognition using isolated signs data and deep transfer learning. *Journal of Ambient Intelligence and Humanized Computing, 14(3),* 1531–1542. https://doi.org/10.1007/s12652-021-03418-z
182. Jiang, S., Liu, Y., Jia, H., Lin, P., He, Z. et al. (2023). Research on end-to-end continuous sign language sentence recognition based on transformer. *2023 15th International Conference on Computer Research and Development (ICCRD)*, pp. 220–226. Hangzhou, China. https://doi.org/10.1109/ICCRD56364.2023.10080216
183. Hu, J., Liu, Y., Lam, K. M., Lou, P. (2023). STFE-Net: A spatial-temporal feature extraction network for continuous sign language translation. *IEEE Access, 11,* 46204–46217. https://doi.org/10.1109/ACCESS.2023.3234743
184. Guo, Q., Zhang, S., Li, H. (2023). Continuous sign language recognition based on spatial-temporal graph attention network. *Computer Modeling in Engineering & Sciences, 134(3),* 1653–1670. https://doi.org/10.32604/cmes.2022.021784
185. Hu, H., Zhao, W., Zhou, W., Li, H. (2023). SignBERT+: Hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9),* 11221–11239. https://doi.org/10.1109/TPAMI.2023.3269220
186. da Silva, D. R. B., de Araujo, T. M. U., do Rego, T. G., Brandao, M. A. C., Goncalves, L. M. G. (2023). A multiple stream architecture for the recognition of signs in Brazilian sign language in the context of health. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-16332-7
187. AlKhuraym, B. Y., Ben Ismail, M. M., Bchir, O. (2022). Arabic sign language recognition using lightweight CNN-based architecture. *International Journal of Advanced Computer Science and Applications, 13(4),* 319–328.
188. Amrutha, K., Prabu, P., Poonia, R. C. (2023). LiST: A lightweight framework for continuous Indian sign language translation. *Information, 14(2),* 79. https://doi.org/10.3390/info14020079

189. Sun, S., Han, L., Wei, J., Hao, H., Huang, J. et al. (2023). ShuffleNetv2-YOLOv3: A real-time recognition method of static sign language based on a lightweight network. *Signal Image and Video Processing, 17(6),* 2721–2729. https://doi.org/10.1007/s11760-023-02489-z

190. Wang, F., Zhang, L., Yan, H., Han, S. (2023). TIM-SLR: A lightweight network for video isolated sign language recognition. *Neural Computing & Applications, 35(30),* 22265–22280. https://doi.org/10.1007/s00521-023-08873-7