Deep Learning–Based Indian Sign Language Recognition: A Comparative Study of DenseNet121 and InceptionV3

AUTHORS: Manju Bhardwaj[1], Shikha Badhani[2], Meenu Mohil[3], Shweta Sankhwar[4], Shireen[5], Harveen[6]

AFFILIATIONS:

[1]Associate Professor, Department of Computer Science. Maitreyi College, University of Delhi, Delhi, India, mbhardwaj@maitreyi.du.ac.in

[2]Associate Professor, Department of Computer Science. Maitreyi College, University of Delhi, Delhi, India, sbadhani@maitreyi.du.ac.in

[3]Associate Professor, Department of Physics, Acharya Narendra Dev College, University of Delhi, Delhi, India, meenumohil@andc.du.ac.in

[4]Assistant Professor, Department of Computer Science. Maitreyi College, University of Delhi, Delhi, India, ssankhwar@maitreyi.du.ac.in

[5]Student, Department of Computer Science. Maitreyi College, University of Delhi, Delhi, India, shireengirotra0009@gmail.com

[6]Student, Department of Computer Science. Maitreyi College, University of Delhi, Delhi, India, kaur06harveen@gmail.com

Abstract

Communication is an integral part of today's world for day-to-day activities. The most prevalent form of communication is through spoken language. However, this mode of communication is not accessible to everyone. In India alone, millions of people are suffering from various communication impairments that restrict an individual's capability to communicate effectively. A person with a communication impairment uses sign language for communication, and this language develops naturally among individuals with similar impairments. However, sign language is not commonly understood by typically abled individuals, leading to a communication gap between the two groups. Sign language serves as a vital means of interaction, yet existing systems often lack accuracy, robustness, or context-specific adaptation, particularly for Indian Sign Language. This research presents two convolutional neural network (CNN) architectures —DenseNet and InceptionV3 for

automated ISL gesture recognition. The proposed models are trained and evaluated on a publicly available Kaggle dataset comprising ISL gesture images, which reflects real-world variations in lighting, background, and signing styles. Experimental results using five-fold cross-validation on the ISL dataset demonstrate that while DenseNet121 consistently outperformed InceptionV3 across all experimental configurations, achieving the highest accuracy of 99.06%. InceptionV3 attained a maximum accuracy of 96.72%, indicating competitive but comparatively lower performance. Extensive evaluation demonstrates that these models can reliably interpret ISL gestures, facilitating real-time translation and fostering social inclusion. This work underscores the potential of advanced deep learning techniques to develop accessible, efficient sign language recognition systems that help bridge communication. gaps and promote social inclusion.

Keywords: Computer vision; DenseNet; Image processing; InceptionV3; Sign language recognition; Deep learning

Introduction

Communication is an essential activity so that people can express their feelings and thoughts, cooperate, and contribute to the general development of society. Effective communication between typically abled individuals and hearing or speech-impaired individuals is a critical challenge across the world, and observed not only in daily communication but in various domains like education and healthcare [1,2].

In India alone, a substantial number of people suffer from various communication impairments that restrict an individual's capability to communicate, thus leaving them in a marginalized section of society [3]. This is where sign language comes into play as a non-verbal communication for people with impairment, but its effectiveness is constrained as not many are aware of interpreting it. Also, in regions like India, this issue is aggravated owing to diverse sign languages [4]. Usually, the sign language translation is done by specially trained human interpreters, but they are rarely available and may be inaccessible many-a-times [1,2].

Sign Language Recognition (SLR) is an essential multidisciplinary research area utilizing concepts like computer vision, pattern recognition and machine learning. Artificial intelligence (AI) and deep learning techniques have been extensively employed for development of SLR systems, which are capable of identifying and translating gestures into text easily [2]. Particularly, Convolutional Neural Networks (CNNs) have been quite successful in visual pattern recognition tasks, thus making them an ideal choice for implementing SLR

systems [5]. These systems have empowered the differently-abled community by automating the gesture interpretation thus reducing their dependence on human interpreters [6].

Deep learning techniques have been widely applied to sign languages such as American, German, Chinese, and Flemish [7,8]. However, comparatively fewer studies have focused on ISL, particularly with robust deep learning models evaluated under diverse and realistic conditions. The signs and gestures used in ISL are quite different from other sign languages, hence, there is a need to develop models specific to this sign language [4]. To address this gap, this research work focuses on developing a robust recognition system, which can handle images in real-world environments with diverse lighting, backgrounds, and signing styles.

This research proposes an AI-based mechanism to translate sign language gestures into text to facilitate communication between sign language and those who are unfamiliar with sign language. This work investigates the applicability of CNN architectures such as DenseNet, and InceptionV3 to identify ISL gestures from the real-world images. By using a balanced dataset, we aim to develop a robust and accurate SLR system. By creating this AI-based communication bridge, we intend to enhance accessibility and inclusivity for speech and hearing-impaired individuals, fostering a more connected and understanding society.

Literature Review

Sign language recognition has attracted significant research interest over the past decade, with numerous approaches utilizing different machine learning and deep learning techniques. Early works primarily focused on handcrafted features, such as shape, color, and motion, which were used to train classifiers like Support Vector Machines (SVMs) and Hidden Markov Models (HMMs) [9,10]. However, these methods often struggled with variability in signing styles, background noise, and lighting conditions.

The advent of deep learning revolutionized the field, introducing convolutional neural networks (CNNs) that automatically learn hierarchical features from raw image data [6]. Several studies demonstrated the effectiveness of CNNs for sign language recognition, especially for American Sign Language (ASL) [11,12]. For instance, Pigou et al. proposed a CNN-based approach that achieved promising results on benchmark datasets, emphasizing the importance of data augmentation for improved generalization [6].

DenseNet architectures, proposed by Huang, Liu, Van der Maaten, and Weinberger (2017) [13], provided enhanced feature reuse through dense connections, leading to more compact and efficient models. Inception models, developed by Szegedy et al., introduced multi-scale feature extraction, which proved effective for recognizing complex gestures and sign sequences [14]. Researchers also explored sequence modeling techniques, combining CNNs with Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units to capture temporal dynamics in sign language videos [15,16].

Despite these advancements, most research has primarily concentrated on Western sign languages, like American Sign Language (ASL) [10, 11, 12] and British Sign Language (BSL) [17].

A comparative study of deep learning approaches for sign language translation, specifically focusing on Inception and Xception architectures is presented in [18]. The study builds upon previous work with VGG19 and MobileNet models to evaluate performance across multiple CNN architectures. Experiments conducted on a dataset of 29,000 sign language images (75×75 pixels) demonstrate exceptional performance, with Inception achieving (99.32 percent) accuracy and Xception reaching (99.43%) accuracy. The results indicate significant potential for CNN-based approaches in sign language recognition systems, with Xception slightly outperforming other tested architectures. In [19], InceptionV3 algorithm yielded favorable results in the domain of sign language gesture recognition. The algorithm gave an accuracy of 98.94%, much better as compared to previous other approaches.

Another work on generation of a real-time system for recognizing Indian sign language uses VGG16 convolutional neural network enhanced with attention mechanism [20]. This study employs a dataset consisting of 23 ISL classes (all alphabets excluding H, J, and Y). Adding attention mechanism to VGG16 enhances accuracy from 97.5% to 99.8%. Also, VGG16 outperforms ResNet50, InceptionV3 and MobileNetV2 for the selected dataset.

Recent studies emphasize the importance of large, diverse datasets and data augmentation techniques to improve model robustness [2, 21]. Transfer learning using pretrained models such as DenseNet, and InceptionV3 has shown promising results in compensating for limited datasets [22]. However, applying these models specifically to ISL remains an ongoing challenge due to the scarcity of annotated datasets and variability in gesture execution. In [23], the authors propose a transfer learning-fused InceptionV3 model to classify murals by dynasty. The model uses InceptionV3 for pretraining on ImageNet, fuses it with transfer learning for parameter readjustment, and extracts deep-level features. The results show the model achieves high accuracy, recall rate, and F1 value, demonstrating its effectiveness in mural classification, thus supporting the use of InceptionV3 for image classification.

Limited work has focused on ISL, which comprises unique gestures and syntactic structures [4]. Many ISL recognition systems tend to rely on traditional image processing and shallow classifiers, yielding limited accuracy under real-world conditions [4, 24].

Recent research continues to focus on developing more sophisticated deep learning models and dataset collection strategies to bridge the gap. Our work aims to contribute by applying advanced CNN architectures to recognize ISL gestures with high accuracy, considering environmental variations and real-world conditions.

Material and Methodology

This section provides details of systematic flow followed to achieve the objectives of our research. The aim of this study is to comparatively analyze the deep CNN models and identify their suitability for designing an Indian sign language recognition system.

Dataset Used

A publicly available Indian Sign Language (ISL) datasets from Kaggle [25] was used in this study, which comprises images of various hand gestures representing alphabets or digits in ISL. The dataset comprises a balanced collection of sign language gestures representing both numerical digits (0-9) and alphabets (A-Z). The images feature diverse real-world conditions to ensure model robustness, including variable lighting conditions, multiple hand orientations and angles, various skin tones representing diversity among signers, different distances between the camera and the signer. For each character, 100 images were randomly selected from the Kaggle dataset. Thus, the dataset comprising 3600 images in total, 100 images each per character (alphabets A-Z, and digits 0-9) was used for deep learning. The dataset includes images captured under diverse lighting conditions and from multiple users. Image augmentation (flipping, rotation, zoom, and contrast adjustment) was applied only to the training set in each cross-validation fold to improve generalization.

Figure 1 displays a sample image of ISL gesture for alphabet 'V' from the selected dataset.

Figure 1: Sample image of ISL gesture from the dataset

Techniques Used

This research focuses on the use of deep learning techniques to develop a robust ISL recognition system. Convolutional neural networks (CNNs) have been found to be quite efficient for extracting features from visual data [5]. This study explores two prominent CNN architectures: DenseNet, and InceptionV3, each technique offering a unique advantage in feature extraction and model efficiency.

1. DenseNet (Dense Convolutional Network), proposed by Huang et al. (2017) [13], emphasizes feature reuse by establishing dense connectivity among layers. Each layer receives inputs from all preceding layers, which encourages the flow of rich gradients and mitigates the vanishing gradient problem further. DenseNet models tend to be more parameter-efficient and have demonstrated excellent performance in image classification tasks.

   In traditional CNNs, the output of the *l-th* layer is:

   $$x_l = H_l(x_l - 1), \quad (1)$$

   where H_l is a non-linear transformation (typically convolution, normalization,

activation). In DenseNet, the *l-th* layer receives feature maps from all preceding layers:

$$x_l = H_l([x_0, x_1, \ldots\ldots, x_{(l-1)}]), \qquad (2)$$

where $([x_0, x_1, \ldots\ldots, x_{(l-1)}])$ represents the concatenation of feature maps produced by layers *0* to *l-1*.

DenseNet121 is a type of deep neural network with 121 layers. It efficiently addresses the vanishing gradient problem often encountered in standard CNNs due to long paths between input and output layers. The architecture consists of four dense blocks containing 6, 12, 24, and 16 convolutional layers respectively. Within these dense blocks, each layer includes batch normalization, ReLU activation, 3x3 convolutions, and dropout. Layers within a dense block are interconnected, enabling each layer to add features to existing feature maps, which are concatenated to retain all learned information. For successful concatenation, feature maps must maintain the same size. Since CNNs often reduce the size through downsampling, this becomes a challenge. DenseNet121 overcomes this by including transition layers between dense blocks, where downsampling occurs to match the sizes. The network produces the final output with a softmax activation function.

DenseNet121 offers several advantages: it interconnects layers to collect diverse feature maps, mitigates vanishing gradients by enabling error detection in early layers, strengthens feature propagation by making early-layer features directly accessible to later layers, and promotes feature reuse, leading to more efficient and effective learning.

2. InceptionV3 takes a different approach by using Inception modules, which perform multiple types of convolutions (1x1, 3x3, 5x5) in parallel and then concatenate their outputs. This allows the network to capture information at multiple scales, improving its ability to learn diverse visual patterns.

   The InceptionV3 architecture, developed in 2015 by Szegedy et al., is a significant advancement over its predecessor models [14]. This architecture is built upon the original GoogLeNet/ Inception design and boasts a depth of 42 layers, incorporating roughly 24 million parameters. The inception modules perform convolution and pooling at multiple scales, capturing diverse features pertinent to gesture recognition. InceptionV3 performs factorized convolutions, which replace larger convolutional operations with smaller ones. For example, a 7x7 convolution can be effectively decomposed into two sequential operations: a 1x7 convolution followed by a 7x1 convolution. This approach drastically reduces the number of parameters while preserving the effectiveness of the model. Additionally, asymmetric convolutions are employed, utilizing different kernel sizes for width and height, which facilitates capturing features at various scales, thereby enhancing efficiency compared to

traditional symmetric convolutions. Another innovative feature of InceptionV3 is the use of auxiliary classifiers situated in the middle of the network. These classifiers help to mitigate the vanishing gradient problem and serve as regularizers during training, thus strengthening the learning process.

InceptionV3 employs grid size reduction to decrease the size of feature maps, achieved through a combination of parallel strided convolutions and max-pooling, which helps maintain computational efficiency. The architecture is also characterized by its inception modules, which consist of multiple parallel paths featuring different convolution sizes. Each path processes the input distinctively, allowing the model to capture many features. Among its notable features, InceptionV3 employs label smoothing, a regularization technique designed to prevent the model from exhibiting excessive confidence in its predictions, ultimately enhancing generalization. Furthermore, batch normalization is applied across all convolution layers, accelerating the training process and reducing internal covariate shifts. The performance benefits of InceptionV3 are substantial.

The method achieves superior computational efficiency due to a reduced parameter count compared to similar architectures, making it more effective in utilizing computing resources and allowing faster training and inference. With respect to accuracy, it boasts a top-5 error rate of 3.58, outperforming previous versions of Inception and positioning itself competitively among other state-of-the-art models. In practical applications, InceptionV3 is primarily used for image classification, feature extraction, transfer learning, and object detection when used as a backbone.

In this work, DenseNet121 and InceptionV3 are utilized with transfer learning, allowing the model to leverage previously learned representations to quickly adapt to the ISL gesture dataset.

Experimental Setup

All the experiments were executed on a laptop with AMD Ryzen™ 7 5800U processor and 16 GB RAM. The project environment was built using Python3 and key libraries including TensorFlow/ Keras for model development, NumPy and Pandas for data handling.

Five-fold cross-validation was performed using KFold from scikit-learn library. In each fold, the data was split into training and validation sets. Augmentation was performed for images in each fold's training set and the CNN models were trained and evaluated on the corresponding validation set. Transfer learning was implemented by freezing the pre-trained convolutional layers and training only the custom classification head while keeping the base model fixed throughout training. The model was trained using the fit function with the training data. This process was performed for 10, 20 and 30 epochs, with a batch size of 16, 32 and 64 each. Callbacks were used to enhance

training by performing actions at specific stages of the training. EarlyStopping (patience = 3) is implemented to monitor validation loss and stop training if no improvement was seen for three consecutive epochs and ModelCheckpoint saves the model with the best validation accuracy during each fold of cross validation. Dropout is applied in the classification head to reduce overfitting.

The experimental implementation employed DenseNet121 and InceptionV3 architectures with ImageNet pre-trained weights, each enhanced with a custom classification head comprising Global Average Pooling (GAP), a fully connected layer (512 units for DenseNet121 and 256 units for InceptionV3, both with ReLU activation), dropout regularization (rate = 0.5), and a softmax output layer for multi-class classification. The models processed RGB images resized to 224 × 224 pixels for DenseNet121 and 299 × 299 pixels for InceptionV3. Experiments were conducted using batch sizes of 16, 32, and 64 across 10, 20, and 30 training epochs. The Adam optimizer (learning rate = 0.001) and categorical cross-entropy loss function were employed for model optimization. Prior to training, images were augmented using random horizontal flipping, rotation (±20°), zoom (10%), and contrast adjustment (20%) to improve model generalization.

The experimental results were quantified through per-fold validation metrics - accuracy, precision, recall, and F1-score, across folds with standard deviation.

Results and Discussion

In this section, we present and analyse the five-fold cross-validation results for obtained on the ISL dataset using the two prominent CNN architectures - DenseNet121 and InceptionV3.

Performance Analysis

Tables 1 and 2 present the five-fold cross-validation results of DenseNet121 and InceptionV3 on the ISL dataset using different combinations of training epochs and batch sizes. Overall, both models showed improved performance with increasing training epochs.

DenseNet121 consistently achieved higher performance than InceptionV3 across all evaluation metrics. Its best results were obtained with 30 epochs and a batch size of 32, achieving an accuracy of 99.06%, precision of 99.12%, recall of 99.06%, and F1-score of 99.04%.

InceptionV3 also exhibited steady improvements as the number of training epochs increased. The best performance was again achieved with 30 epochs and a batch size of 32, reaching an accuracy of 96.72%, precision of 97.02%, recall of 96.72%, and F1-score of 96.68%.

Overall, the results indicate that increasing the number of training epochs enhanced the performance of both models. However, DenseNet121 consistently outperformed InceptionV3, achieving the highest classification performance on the ISL dataset and demonstrating its suitability for the proposed recognition task.

Table 1. Five-fold cross validation results obtained on ISL dataset using DenseNet121

Epochs	Batch size	Accuracy	Precision	Recall	F1-Score
10	16	0.9644 ± 0.0115	0.9681 ± 0.0096	0.9644 ± 0.0115	0.9644 ± 0.0114
	32	0.9664 ± 0.0112	0.9702 ± 0.0096	0.9664 ± 0.0112	0.9659 ± 0.0116
	64	0.9539 ± 0.0126	0.9592 ± 0.0110	0.9539 ± 0.0126	0.9532 ± 0.0128
20	16	0.9808 ± 0.0040	0.9829 ± 0.0034	0.9808 ± 0.0040	0.9806 ± 0.0039
	32	0.9858 ± 0.0043	0.9871 ± 0.0042	0.9858 ± 0.0043	0.9856 ± 0.0043
	64	0.9806 ± 0.0088	0.9819 ± 0.0086	0.9806 ± 0.0088	0.9801 ± 0.009
30	16	0.9883 ± 0.0067	0.9898 ± 0.0058	0.9883 ± 0.0067	0.9883 ± 0.0067
	32	**0.9906 ± 0.0054**	**0.9912 ± 0.0052**	**0.9906 ± 0.0054**	**0.9904 ± 0.0055**
	64	0.9877 ± 0.0063	0.9889 ± 0.0055	0.9877 ± 0.0063	0.9875 ± 0.0064

Table 2. Five-fold cross validation results obtained on ISL dataset using InceptionV3

Epochs	Batch size	Accuracy	Precision	Recall	F1-Score
10	16	0.8858 ± 0.0133	0.9025 ± 0.0202	0.8858 ± 0.0133	0.8815 ± 0.0151
	32	0.8953 ± 0.0155	0.9135 ± 0.0117	0.8953 ± 0.0155	0.8942 ± 0.0163
	64	0.9083 ± 0.0118	0.9225 ± 0.0089	0.9083 ± 0.0118	0.9068 ± 0.0127
20	16	0.9294 ± 0.0166	0.9374 ± 0.0139	0.9294 ± 0.0166	0.9286 ± 0.0168
	32	0.9486 ± 0.0170	0.9541 ± 0.0140	0.9486 ± 0.0170	0.9479 ± 0.0173
	64	0.9531 ± 0.0119	0.9596 ± 0.0068	0.9531 ± 0.0119	0.9520 ± 0.0128
30	16	0.9525 ± 0.0101	0.9578 ± 0.0079	0.9525 ± 0.0101	0.9522 ± 0.0097
	32	**0.9672 ± 0.0043**	0.9702 ± 0.0040	**0.9672 ± 0.0043**	**0.9668 ± 0.0046**

	64	**0.9672 ± 0.0157**	**0.9705 ± 0.0129**	**0.9672 ± 0.0157**	0.9667 ± 0.0159

Discrimination Analysis

To further evaluate the two approaches, we examine the ability of DenseNet121 and InceptionV3 to discriminate between visually similar gestures. Figures 2 and 3 present the confusion matrices (Epochs = 30, Batch Size = 32) corresponding to the best performance of DenseNet121 and InceptionV3 respectively. In

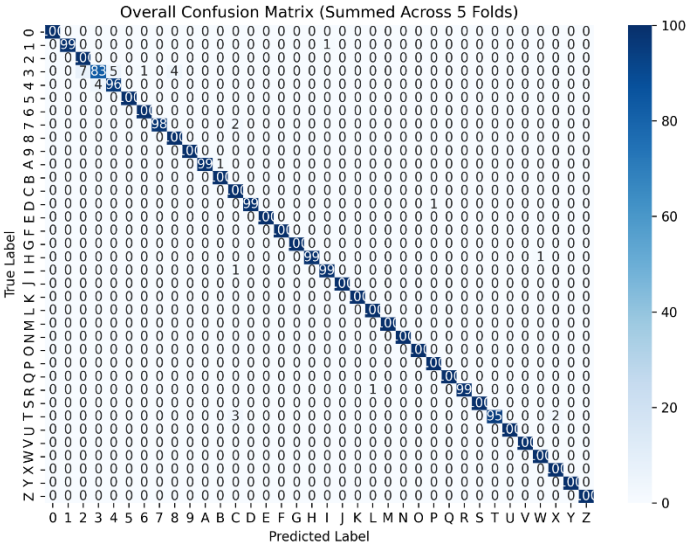

Figure 2. Confusion Matrix for ISL Dataset using DenseNet121

both cases, most of the predictions line up neatly along the diagonal, which means the models correctly predict the true class most of the time. The diagonal values are close to 100, showing that both are very good at recognizing digits and letters. Misclassifications are rare, with only a handful of off-diagonal entries.

DenseNet121 is remarkably consistent, with diagonal values ranging from 98 to 100. It almost never struggles, though there are a few exceptions: the digit *3* sometimes gets confused with *2*, *4*, or *8*, and the letter *T* occasionally slips into *C*. InceptionV3, on the other hand, shows a wider range of accuracy (88 to 100). It has more trouble with digits, especially those between *0* and *5*. For letters, the biggest mix-ups are *D* being mistaken for *P* and *M* for *H*. These errors make sense when you consider the similarity in ISL gestures for those characters — the shapes are close enough that the model occasionally gets tripped up.

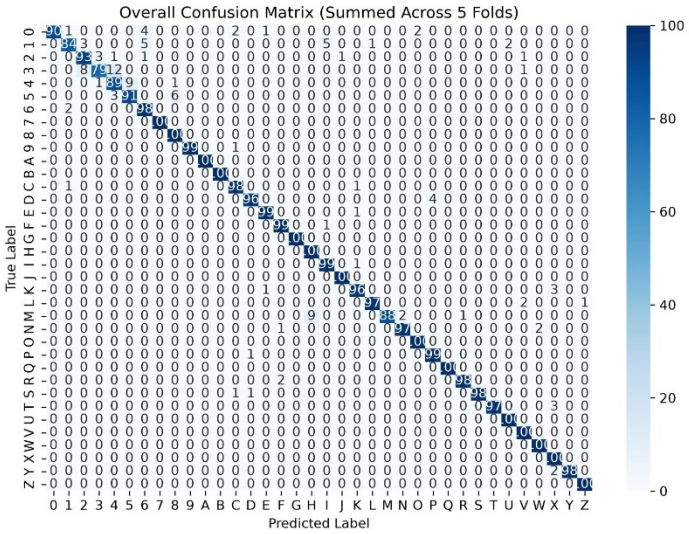

Figure 3. Confusion Matrix for ISL Dataset using InceptionV3

Thus, DenseNet121 delivers steadier performance, while InceptionV3 is good but more prone to confusion when gestures are visually alike.

References

1. Leeson L, Wurm S, Vermeerbergen M, editors. Signed language interpreting:

Preparation, practice and performance. London: Routledge; 2011.

2. Tao T, Zhao Y, Liu T, Zhu J. Sign language recognition: a comprehensive review of traditional and deep learning approaches, datasets, and challenges. IEEE Access. 2024;12:75034-75060.
3. Office of the Registrar General & Census Commissioner, India. Census of India 2011: Data on Disability [Internet]. New Delhi: Government of India; 2011 [cited 2026 Jun 18]. Available from: https://unstats.un.org/unsd/demographic-social/meetings/2016/bangkok--disability-measurement-and-statistics/Session-6/India.pdf
4. Damdoo R, Kumar P. An integrative survey on Indian sign language recognition and translation. IET Image Process. 2025;19(1):e70000.
5. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc IEEE Conf Comput Vis Pattern Recognit (CVPR); 2016. p. 770-778. doi:10.1109/CVPR.2016.90.
6. Pigou L, Dieleman S, Kindermans PJ, Schrauwen B. Sign language recognition using convolutional neural networks. In: Agapito L, Bronstein MM, Rother C, editors. Computer Vision – ECCV 2014 Workshops. Lecture Notes in Computer Science. Vol. 8925. Cham: Springer; 2015. p. 572-578. doi:10.1007/978-3-319-16178-5_40.
7. Ananthanarayana T, Srivastava P, Chintha A, Santha A, Landy B, Panaro J, Webster A, Kotecha N, Sah S, Sarchet T, Ptucha R, Nwogu I. Deep learning methods for sign language translation. ACM Trans Access Comput. 2021;14(4):22. doi:10.1145/3477498.
8. Martinez-Martin E, Morillas-Espejo F. Deep learning techniques for Spanish sign language interpretation. Comput Intell Neurosci. 2021;2021:5532580. doi:10.1155/2021/5532580.
9. Al Abdullah BA, Amoudi GA, Alghamdi HS. Advancements in sign language recognition: a comprehensive review and future prospects. IEEE Access. 2024;12:128871-128895. doi:10.1109/ACCESS.2024.3457692.
10. Starner T, Weaver J, Pentland A. Real-time American Sign Language recognition using desk and wearable computer-based video. IEEE Trans Pattern Anal Mach Intell. 1998;20(12):1371-1375. doi:10.1109/34.735811.
11. Sharma S, Kumar K. ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. Multimed Tools Appl. 2021;80:26319-26331. doi:10.1007/s11042-021-10768-5.
12. Lee YB, Goh YH, Lum KY. Study of convolutional neural network in recognizing static American Sign Language. In: Proc IEEE Int Conf Signal Image Process Appl (ICSIPA); 2019. p. 41-45. doi:10.1109/ICSIPA45851.2019.8977767.
13. Huang G, Liu Z, Van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proc IEEE Conf Comput Vis Pattern Recognit (CVPR); 2017. p. 2261-2269. doi:10.1109/CVPR.2017.243.
14. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proc

IEEE Conf Comput Vis Pattern Recognit (CVPR); 2016. p. 2818-2826. doi:10.1109/CVPR.2016.308.

15. Adaloglou N, Chatzis T, Papastratis I, Stergioulas A, Papadopoulos GT, Zacharopoulou V, et al. A comprehensive study on deep learning-based methods for sign language recognition. IEEE Trans Multimed. 2021;24:1750-1762.
16. Myagila K, Nyambo DG, Dida MA. Efficient spatio-temporal modeling for sign language recognition using CNN and RNN architectures. Front Artif Intell. 2025;8:1630743.
17. Hameed H, Usman M, Tahir A, Ahmad K, Hussain A, Imran MA, Abbasi QH. Recognizing British sign language using deep learning: a contactless and privacy-preserving approach. IEEE Trans Comput Soc Syst. 2023;10(4):2090-2098.
18. Abu-Jamie TN, Abu-Naser SS. Classification of sign-language using deep learning: a comparison between Inception and Xception models. Int J Acad Eng Res. 2022;6(8):9-19.
19. Kankariya S, Thakre K, Solanki U, Mali S, Chunawale A. Sign language gestures recognition using CNN and Inception V3. In: Proc Int Conf Emerg Smart Comput Inform (ESCI); 2024. p. 1-6.
20. Kumar S, Rani R, Chaudhari U. Real-time sign language detection: empowering the disabled community. MethodsX. 2024;13: 102901
21. Singla V, Bawa S, Singh J. Enhancing Indian sign language recognition through data augmentation and visual transformer. Neural Comput Appl. 2024;36(24):15103-15116.
22. Alharthi NM, Alzahrani SM. Vision transformers and transfer learning approaches for Arabic sign language recognition. Appl Sci. 2023;13(21):11625.
23. Cao J, Yan M, Jia Y, Tian X, Zhang Z. Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals. EURASIP J Adv Signal Process. 2021;2021(1):49.
24. Sharma S, Singh S. Recognition of Indian sign language (ISL) using deep learning model. Wirel Pers Commun. 2022;123(1):671-692.
25. Kaggle. Indian sign language dataset [Internet]. 2024 [cited 2026 Mar 31]. Available from: https://www.kaggle.com/datasets/atharvadumbre/indian-sign-language-islrtc-referred.