Advances in Artificial Intelligence and Machine Learning; Research 6 (4) 5983-5994 Received 29-04-2026; Accepted 07-08-2026; Published online 14-08-2026

A Real-Time Deep Learning-Based Sign Language Recognition System for Words, Alphabets, and Numbers

Milia Habib milia.habib@liu.edu.lb
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Teddy Nohra tnohra91@gmail.com
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Charbel Srour Charbelsrourr@gmail.com
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Mohammad Kanaan mohamad.kanaan01@liu.edu.lb
Department of Electrical Engineering
Lebanese International University
Beirut, Lebanon

Zaher Merhi zaher.merhi@liu.edu.lb
Department of Computer & Communications Engineering
Lebanese International University
Beirut, Lebanon

Corresponding Author: Milia Habib

Copyright © 2026 Milia Habib et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

This paper presents a sign language recognition system based on deep learning and computer vision. It aims to support communication between deaf and hard-of-hearing individuals and the community. The proposed system translates hand gestures into textual output through real-time image and video processing. It supports three recognition modes, including number, alphabet, and word recognition. For alphabet and digit recognition, MobileNetV2 is adopted due to its lightweight architecture and suitability for real-time deployment. To improve recognition accuracy and robustness in practical scenarios, a custom-collected dataset is combined with publicly available American Sign Language (ASL) datasets. The resulting dataset covers static alphabet gestures (excluding the motion-based letters J and Z) and digit gestures corresponding to the numbers 0–9. For word recognition, the pretrained Inflated 3D ConvNet (I3D) model is adopted to process short video clips and learn both spatial and temporal gesture features. Additionally, OpenCV is employed for video processing and real-

Citation: Milia Habib et al. A Real-Time Deep Learning-Based Sign Language Recognition System for Words, Alphabets, and Numbers. **Advances in Artificial Intelligence and Machine Learning**. 2026;6(4):331. https://dx.doi.org/10.54364/AAIML.2026.64331

time frame capture, while DroidCam is used to capture live input. For the MobileNetV2 model, a Region of Interest (ROI) is applied before prediction. Experimental results from real-time testing demonstrate high confidence scores for alphabet and number recognition, achieving 93.43% and 92.8%, respectively, at 30 FPS. In addition, the word recognition module achieves moderate performance, with Top-1 and Top-5 accuracies of 39.2% and 68.0%, respectively.

Keywords: Sign language recognition, Deep learning, MobileNetV2, I3D, Real-Time gesture recognition, OpenCV.

1. INTRODUCTION

Communication is vital to daily human life. It is an exchange of ideas, emotions, and information. For a deaf or hard-of-hearing person, sign language is their main way to communicate with the public. However, many people do not understand sign language, which often creates a communication barrier between deaf people and the community. According to the World Health Organization (WHO), more than 430 million people worldwide—approximately 5% of the global population—live with disabling hearing loss, and this number is expected to reach over 700 million by 2050 [1]. In recent years, communication between deaf people and hearing-impaired and non-hearing individuals has become increasingly important, especially in public services, education, and daily interaction. Sign language is the primary mode of communication for the deaf community today. However, the absence of understanding of sign language among the general population creates a significant communication problem. This barrier often results in misunderstandings, difficulties, and limited access to essential services. The development and evaluation of sign language recognition systems have become an important research area in computer vision and artificial intelligence. Deep learning models, large datasets, and cameras have enabled the recording of hand gestures and movements. A system can now translate these gestures into textual output. The SLR system (Sign Language Recognition) combines images that are preprocessed with deep learning techniques to recognize gestures through hand movements. For SLR system, it was valued around 500 million USD in 2025, with a projection that shows the annual growth due to the increasing use of communication tools for accessible AI-driven [2]. With all this progress in sign language detection, there is still a difficulty. Each hand gesture can be recognized depending on some conditions like lighting, colors, hand orientation, and background complexity [3]. As a result, building an efficient and reliable SLR system from a live video capture is challenging. To handle these problems, this research focuses on developing a real-time inference sign language recognition system using a deep learning approach for three modes: numbers, alphabet, and words.

2. RELATED WORK

Sign language recognition has become a prominent research area, particularly with the rapid growth in computer and machine vision and artificial intelligence. Tolentino et al. (2019) [4], presented a vision-based system designed to recognize static ASL hand signs using deep learning techniques. The authors limited their work to static hand postures such as letters, numbers, and basic signs, since these can be processed directly by a Convolutional Neural Network (CNN), which operates

on single images rather than motion sequences. The system was intended to function as a simple and accessible learning tool for beginners who want to practice ASL hand signs using only a webcam and a deep learning classifier. The proposed system achieved an average testing accuracy of 93.67%, including 90.04% for ASL alphabet recognition, 93.44% for number recognition, and 97.52% for static word recognition. The researchers in [5] presented a deep learning model, CNNSa-LSTM. The prototype uses VGG16 to pull out features from samples and extract motion information. It combines the 2 models, CNN and LSTM components. It reached an accuracy of 98.7%, a sensitivity of 98.2%, a precision of 98.5%, a WER of 0.131, a SER of 0.114, and an NDCG of 98%, proving strong total performance. Garg et al. (2025) [6], concentrated on automatic sign language detection using CNNs to categorize hand motions. They used a dataset including 26,000 image samples, with 1000 images for each hand alphabet sign. The images were collected under controlled lighting conditions with a black background to improve feature extraction. Participants of different ages, skin tones, and hand shapes were included to enhance dataset diversity. The dataset was divided into 70% training, 15% validation, and 15% testing sets. This model achieved high accuracy. To reduce the communication barrier between deaf or mute individuals and the general public, Dadge et al. (2025) [7], proposed a lightweight hand gesture recognition system for real-time communication. The framework combines MediaPipe hand landmark detection with a machine learning classifier to recognize static and dynamic hand gestures. Only seven gestures were tested, using a dataset of 3500 samples (A, B, C, D, Open, Close, and OK). The system achieved an accuracy of 94.1% at 30 FPS (frames per second). Compared with CNN, Transformer, and TinyML-based models, the proposed approach offers a good balance of accuracy, speed, and real-time accessibility. Arasavali et al. (2024) [8], proposed a low-cost ASL recognition system using CNN on the ”Finger Spelling, A” image dataset containing 24 alphabet letters, excluding J and Z, since these require movement. The dataset includes varied backgrounds, making recognition more challenging and realistic. A deep learning and computer vision approach was also proposed by Kale et al. (2022) [9], for ASL detection. The system employed a Linear SVM algorithm and MediaPipe libraries to translate gesture alphabets using 16 datasets, including A–K and U, V, W, and Y. In addition, a gesture-based communication tool based on a flex sensor, an accelerometer, Arduino, and a WI-FI module is implemented by Pavitra et al. (2023) [10], for Mute and Hearing Impaired People. Harikrishnan et al. (2023) [11], developed a CNN-based system to recognize and convert sign language into text and voice using image data captured by a computer webcam. In this context, and to address these challenges, this work focuses on developing a real-time sign language recognition system with live inference that displays the predicted output directly during hand gesture performance. The system relies on MobileNetV2 and I3D models to recognize alphabets, numbers, and words from live video streams. Unlike many existing studies, the proposed system performs real-time recognition and displays the corresponding textual output directly on the screen, rather than simply reporting the training results.

3. DATASET AND PROPOSED METHODOLOGY

3.1 Data and Preprocessing

In this study, multiple datasets were used to train different recognition models for the sign language recognition system, covering three modes: words, alphabets, and numbers. A pretrained I3D model, originally trained on the WLASL dataset, is used directly for inference in this project [12]. This

dataset contains 2000 classes represented in around 21,000 short videos, and 119 signers participated in recording the signs. These videos were processed by extracting frames, which were later used to train the I3D model. WLASL uses a signer-independent split protocol, where all samples belonging to a specific signer are assigned exclusively to one subset (training, validation, or testing). This prevents overlap of signer identities across splits and enables evaluation on unseen signers. For alphabet recognition, the ASL Alphabet Image Dataset was employed [13]. The dataset contains 3,000 samples for each alphabet letter, represented by different hand gestures captured under various conditions, including diverse backgrounds, lighting settings, hand positions, and hand sizes. To enhance dataset diversity and improve real-time performance, an additional 500 samples per letter were manually collected using a camera from a single participant, resulting in a total of 3,500 samples for each letter. The dataset does not include the letters "J" and "Z" because these signs involve motion and therefore require video sequences rather than static images for accurate representation. For number recognition, a separate ASL digits dataset collected under different conditions was used [14]. This dataset contains 500 samples for each digit. To further improve dataset diversity and real-time performance, an additional 500 samples per digit were manually collected from the same participant and incorporated into the dataset, resulting in 1,000 samples per digit. It should be noted that all additional alphabet and digit samples were collected from a single participant, who also participated in the real-time inference experiments. For the number recognition mode, the dataset was divided into 80% training data and 20% validation data. For the alphabet recognition mode, the dataset was divided into 70% training data, 20% validation data, and 10% testing data. Regarding the preprocessing, We applied the following preprocessing steps to ensure that the data is consistent, suitable, and efficiently processed before starting the training process.

1. Image resizing: All images are resized to 224x224. This ensures that all inputs have the same size before being fed into the MobileNetV2 model.
2. Random Resized Crop: It is applied to randomly crop and resize images, creating variations in hand position within the Region of Interest (ROI). For the alphabet and number recognition modes, the ROI size is 316 × 288 pixels. After extraction, the ROI is resized to 224 × 224 pixels before being used as input to the model.
3. Random Rotation and Random Affine: These transformations are used to simulate variations in hand movement and orientation during real-time conditions.
4. Random Perspective: It is responsible for handling different camera angles.
5. Color Jitter: It is applied to adjust brightness and saturation, enabling the model to handle variations in lighting conditions.
6. Gaussian Blur and Random Adjust Sharpness: They are applied to introduce variations in image quality and sharpness.
7. Tensor the output: This final step is applied to the processed images into tensor format before being passed to the MobileNetV2 model for training.

3.2 Proposed Methodology

In this study, the system used the MobileNetV2 algorithm for alphabet and number recognition [6, 15]. This model contains 53 layers and employs inverted residual blocks with linear bottlenecks to reduce computational complexity and memory usage. FIGURE 1, presents the MobileNetV2 model architecture. It is chosen due to its suitability for real-time classification applications. The system captures a defined image from a region of interest (ROI), which is then preprocessed and passed to

the trained MobileNetV2 model to predict the output. For the word recognition, the system uses the pretrained I3D Inflated 3D ConvNet algorithm [16]. The model receives a video clip of 64 RGB frames. Each frame is resized to 256 pixels and normalized to the range [-1,1] before being passed to the model for inference. FIGURE 2, presents the I3D model architecture. It consists of more than 100 layers and processes short video clips instead of single images. It extends a traditional 2D convolution neural network to 3D by adding a temporal dimension, allowing the network to capture both spatial and temporal information. It is selected due to its capability of learning both spatial and temporal features from video sequences. FIGURE 3, summarizes the system overflow. TABLE 1, lists the training configuration for the alphabet and number MobileNetV2 models.

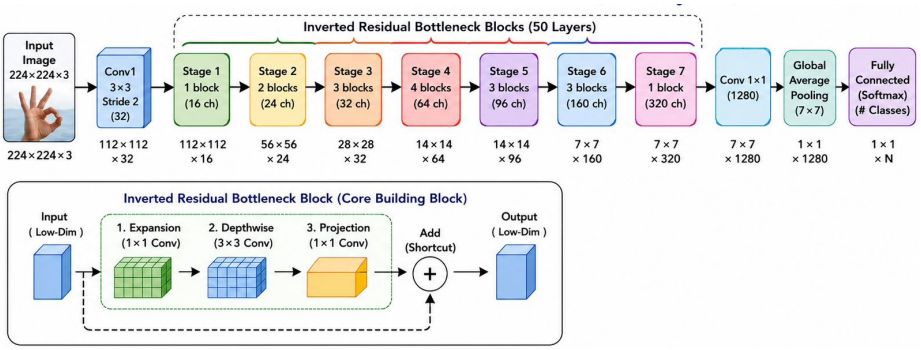

Figure 1: MobileNetV2 architecture.

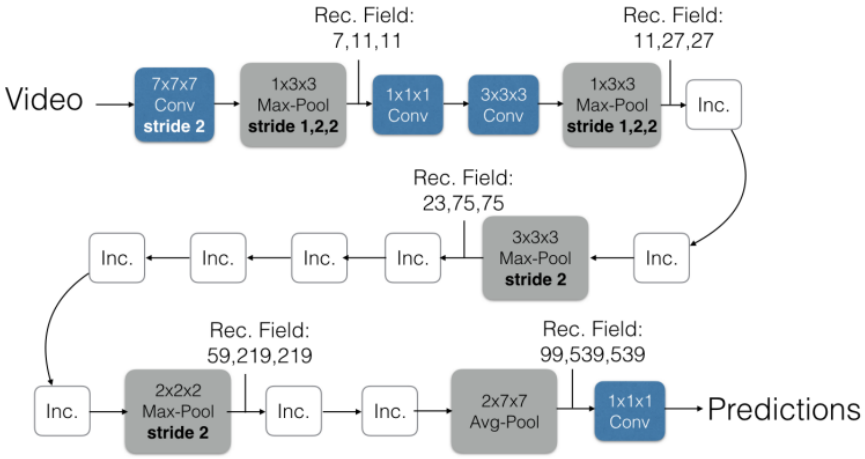

Figure 2: I3D architecture [17].

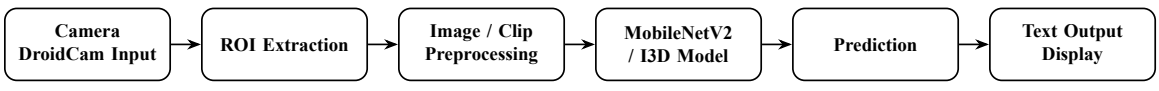

Figure 3: System Workflow

Table 1: Training Configuration of MobileNetV2 Models

Parameter	Alphabet Mode	Number Mode
Model	MobileNetV2	MobileNetV2
Epochs	15	10
Batch Size	32	32
Optimizer	AdamW	AdamW
Learning Rate	1×10^{-4}	1×10^{-4}
Weight Decay	1×10^{-4}	1×10^{-4}
Loss Function	Cross-Entropy Loss	Cross-Entropy Loss
Label Smoothing	0.05	0.05
Dropout	0.35	0.35
Random Seed	42	42

4. MODEL PERFORMACE AND EXPERIMENTALS RESULTS

4.1 Evaluation Metrics

In this study, several metrics were used to evaluate the performance of the real-time sign language recognition system, as listed below. For numbers and alphabet modes, the confidence score is calculated as illustrated in Equation 1.

$$\mathbf{p_i} = \frac{\mathbf{e^{z_i}}}{\sum_{\mathbf{j=1}}^{\mathbf{C}} \mathbf{e^{z_j}}} \tag{1}$$

where $\mathbf{p_i}$ is the confidence of class $\mathbf{i}$, $\mathbf{z_i}$ is its raw score, and $\mathbf{C}$ is the number of classes, which is 24 for alphabet recognition and 10 for number recognition.
For the words mode, the confidence score is shown in Equation 2.

$$s_i = \max_t(z_{i,t}) \tag{2}$$

where $z_{i,t}$ is the raw model score for word class i at time t, s_i is the final score of word class i after temporal max pooling. Softmax is then applied across all word classes as shown in Equation 3.

$$p_i = \frac{e^{s_i}}{\sum_{j=1}^{C} e^{s_j}} \tag{3}$$

where p_i is the word confidence of class i, C is the number of word classes.

Cross Entropy Loss was applied to measure the error between the actual gestures and the predicted outputs as shown in Equation 4.

$$L = -\sum_{i=1}^{C} y_i \log(p_i) \tag{4}$$

where y_i is the actual label, p_i is the predicted probability, C is the number of classes.

Top-1 Accuracy (Equation 5) measures the proportion of test samples for which the model's most confident prediction is correct. .

$$\text{Top-1 Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I(\hat{y}_i^{(1)} = y_i) \tag{5}$$

The Top-5 accuracy metric (Equation 6) considers a prediction correct if the ground truth label appears among the model's five most confident predictions.

$$\text{Top-5 Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I\left(y_i \in \left\{\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \hat{y}_i^{(3)}, \hat{y}_i^{(4)}, \hat{y}_i^{(5)}\right\}\right) \tag{6}$$

4.2 Experimental Results and Discussion

We developed a user-friendly interface to display the system output, as shown in FIGURE 4. The user selects the desired recognition mode and performs hand gestures within the green region, which represents the Region of Interest (ROI). Once a sign is recognized, the predicted alphabet and its corresponding confidence score are displayed. The confidence score shown in the real-time interface represents the predicted class probability generated by the model. The user can then press the ENTER key to append the predicted character to the text area displayed in the black rectangle. For each prediction, the confidence score, predicted character, and selected mode are shown in the upper-left corner of the ROI, together with the processing rate of 30 FPS. Here, 30 FPS refers to the webcam capture and display refresh rate during real-time operation. In word recognition mode, the *Record Word* button allows the user to capture an additional 2.5-second video clip for subsequent recognition. The system acquires live video through DroidCam, extracts the ROI, preprocesses the input, and performs recognition using MobileNetV2 for alphabets and digits and I3D for words. The predicted class is then converted into text and displayed in real time. The experiments were conducted on a Windows machine equipped with a 13th Gen Intel Core i7 processor, 16 GB of RAM, and Intel UHD Graphics with 8 GB of reported graphics memory. The system was implemented in Python, utilizing the PIL/Pillow library for image handling. FIGURE 5 and FIGURE 6, present the output predicted by our system for the number 5 and letter V, respectively.

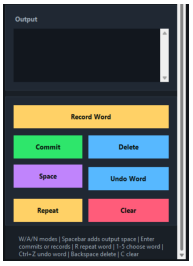

Figure 4: System Interface

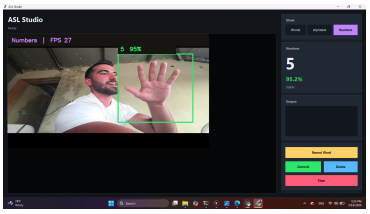

Figure 5: Number 5 gesture

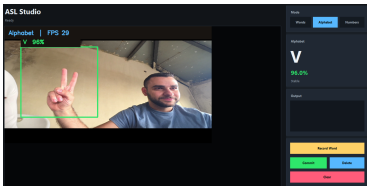

Figure 6: Letter V gesture

We evaluated three recognition modes: alphabets, numbers, and words. The evaluation was conducted under two environmental settings: **In-Dataset**, which used a white-wall background similar to that of the training data, and **Out-of-Dataset**, which involved arbitrary backgrounds and environmental conditions not represented during training. TABLE 2 and TABLE 3, present the confidence scores obtained for the number and alphabet recognition modes during real-time testing, respectively. In these experiments, new gesture samples were captured using DroidCam under varying environmental conditions. The results demonstrate high recognition confidence across all digit classes, achieving an average confidence score of 92.8% under Out-of-Dataset conditions. Under In-Dataset conditions, the average confidence score increased to 96.89%, which is consistent with the model's strong performance during training. Similarly, the alphabet recognition system reached an average confidence score of 93.43% for samples captured under conditions different from those of the training dataset (like hand position, lighting, colors, and background) and an average confidence score of 96.67% for the same dataset conditions. This result indicates the effectiveness of the model in real-time performance. TABLE 4, presents the training and validation loss of the two modes, alphabet and numbers. It indicates good generalization performance and low validation error for both modes.

Table 2: Confidence score (%) of number recognition gestures at 30 FPS.

Number	Out-of-Dataset Conditions	In-Dataset Conditions
0	94.0	96.9
1	92.7	97.3
2	92.5	97.5
3	90.4	96.8
4	93.4	96.5
5	95.2	97.1
6	91.9	96.8
7	96.1	96.1
8	91.7	96.8
9	90.1	97.1
Average	**92.8**	**96.89**

Table 3: Confidence score (%) of alphabet recognition gestures at 30 FPS.

Alphabet	Out-of-Dataset Conditions	In-Dataset Conditions
A	86.1	97.1
B	94.7	96.2
C	93.7	96.4
D	94.5	96.9
E	95.8	96.9
F	94.9	96.7
G	94.1	97.2
H	95.6	97.1
I	94.6	97.8
K	93.6	96.1
L	94.0	96.5
M	95.3	96.3
N	95.3	96.8
O	91.4	95.8
P	93.3	95.7
Q	94.8	96.1
R	93.5	96.3
S	90.8	97.6
T	94.8	97.1
U	94.2	96.9
V	96.0	96.2
W	88.4	96.8
X	90.0	96.3
Y	93.1	97.1
Average	**93.43**	**96.67**

Table 4: Training and Validation Loss for Alphabet and Numbers Recognition Modes

Mode	Training Loss	Validation Loss
Numbers Recognition	0.3320	0.2996
Alphabet Recognition	0.3490	0.3438

For the word recognition system based on the I3D model, the authors in [16] reported a training Top-1 accuracy of 32.48% and a training Top-5 accuracy of 57.31%. To evaluate the system in a real-time scenario, we tested 25 words under real-time inference conditions. Each word was tested five times, and the predicted outputs of the captured short video clips were recorded and analyzed. The evaluation was limited to 25 words due to the small size of the available dataset, which contains only a few samples per word. This data limitation makes correct prediction more challenging for the system. TABLE 5, shows that the average Top-1 accuracy reached 39.2%, while the average Top-5 accuracy reached 68.0%. Some words were correctly predicted as Top-1 in all five attempts, while

Table 5: Top-1 and Top-5 Accuracy at 30 FPS.

Words	Top-1 (%)	Top-5 (%)
Baby	100	100
Arm	80	100
Book	80	100
Chop	80	100
Community	80	100
Laptop	100	100
Pizza	100	100
Scissors	100	100
Yes	0	0
Hello	0	0
Drink	0	60
We	60	80
Love	80	100
Table	20	80
Monkey	20	80
No	0	0
Help	0	40
Stomach	20	40
Brain	20	60
Telephone	40	100
Hug	0	80
If	0	60
You	0	0
Appreciate	0	60
Thanks	0	60
Total Average	**39.2**	**68.0**

others appeared only within the Top-5 predictions or were not predicted correctly at all. These results highlight the challenges of real-time word-level recognition, aligning with the training outcomes reported in [16].

5. CONCLUSION AND FUTURE WORK

In this study, we developed an efficient sign language recognition system using deep learning models to facilitate communication with deaf individuals by translating hand gestures into textual output through a simple interface. The system included three recognition modes: alphabet, number, and word recognition. The MobileNetV2 model was used for alphabet and number recognition, while the I3D model was employed for word recognition. The results demonstrated a useful system of high confidence scores for alphabet and number recognition, with moderate accuracy for

word recognition during real-time inference. These results highlight several challenges concerning dataset conditions and live testing environments, especially with variation in lighting, recording quality, hand positions, and gesture accuracy. As future work, the dataset can be updated and increased by collecting more samples with a wide variety of backgrounds and camera angles. This modification can improve the system's effectiveness in real-time inference. Secondly, this system can be deployed on smart devices or smart glasses to make it more useful for daily use. Finally, testing this system with hard-to-hear and deaf people would give helpful feedback in improving the system interface, accuracy, and serviceability.

Author Contribution

All authors made equal contributions to this work in conceptualization, methodology, formal analysis, writing, reviewing, and editing.

References

[1] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] https://www.datainsightsmarket.com/reports/sign-language-translation-software-1956596

[3] Huang J, Chouvatut V. Video-Based Sign Language Recognition via ResNet and LSTM Network. J Imaging. 2024;10:149.

[4] Tolentino LK, Juan RS, Thio-ac AC, Pamahoy MA, Forteza JR, et al. Static Sign Language Recognition Using Deep Learning. Int J Mach Learn Comput. 2019;9:821-827.

[5] Baihan A, Alutaibi AI, Alshehri M, Sharma SK. Sign Language Recognition Using Modified Deep Learning Network and Hybrid Optimization: A Hybrid Optimizer (Ho) Based Optimized Cnnsa-LSTM Approach. Sci Rep. 2024;14:26111.

[6] Garg B, Kasar M, Paygude P, Dhumane A, Ambala S, et al. Sign Language Detection Dataset: A Resource for Ai-Based Recognition Systems. Data Br. 2025;61:111703.

[7] Dagde R, Thakre S, Chopade S, Rokde L, Kakani V. A Hand Sign Recognition Based Signal System for Mute People Using Machine Learning. MethodsX. 2025;15:103670.

[8] Arasavali N, Aryasomayajula SS, Talari SS, Mounika JS. Sign Language Prediction Using Eight-Layered CNN for Deaf and Dumb. In 2024 International Conference on Electronics, Computing, Communication and Control Technology (ICECCC). IEEE. 2024:1-4.

[9] Kale P, Nemade M, Karia A, Majalkar V. SIGN-UNITE Asl Detection Using ML for Deaf Mute People. In 2022 5th International Conference on Advances in Science and Technology (ICAST). IEEE. 2022:471-475.

[10] Pavitra N, Vijay K, Sharathkumar NS, Punithkumar HM, Nirmala BL, et al. A Gesture Based Communication Tool for Mute and Hearing Impaired People. In 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA). IEEE. 2023:185-190.

[11] Harikrishnan H, Johny J, Mathews MC, Suresh M, Joseph H. Hand Gesture Recognition and App for Deaf and Dumb People. In 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT). IEEE. 2023:777-781.

[12] https://www.kaggle.com/datasets/sttaseen/wlasl2000-resized.

[13] https://www.kaggle.com/datasets/grassknoted/asl-alphabet/data.

[14] https://www.kaggle.com/datasets/rayeed045/american-sign-language-digit-dataset/data.

[15] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MOBILENETV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2018:4510-4520.

[16] Li D, Rodriguez C, Yu X, Li H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. 2019. arXiv preprint: https://arxiv.org/pdf/1910.11006v1.

[17] Carreira J, Zisserman A. Quo Vadis, Action Recognition? A New Model and the Kinetics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition . 2017:4724-4733.