ISSN 2007-9737

Indian Sign Language Recognition using MobileNetV2 Fine-Tuned by Transfer Learning

Sanjit Kumar Dash[1,*], Abhinash Padhi[1], Aditya Kumar Sahu [1], Muktikanta Sahu[2]

[1] Odisha University of Technology and Research, Bhubaneswar,
India

[2] International Institute of Information Technology Bhubaneswar,
India

{abhinashpadhi79, adityasahuaahan123} @gmail.com, skdash@outr.ac.in, muktikanta@iiit-bh.ac.in

Abstract. Sign Language is the language used for communication involving hearing impaired and hearing disabled people that involves the movement of hands to exchange information. But even with the existence of such language, people find it difficult to communicate using the same due to its vast diversity across different regions and geographical areas of the world. For instance, ISL (Indian Sign Language) and ASL are the respective sign languages used in USA and India but they are completely different from one another from the perspective of hand signs as well as understanding. This arises the requirement for a model which provides people a basis to translate and understand ISL.The model that has been used in this work involves a pretrained model, MobileNetV2, which is further aided by fine-tuning and Transfer Learning techniques so that the model's components are reapplied to the new model thereby reducing time and computational resources. The Indian Sign Language (ISLRTC referred) dataset is employed using signs demonstrated on the ISLRTC website taken as images under different lighting conditions and backgrounds and is preprocessed and augmented thereby undergoing operations like Rescaling, Normalization, Standardization of pixels. It consists of 36 labeled classes(26 Alphabets + 10 digits) each containing a set of 1000 sample images that represent a certain gesture. The preprocessed dataset is then splitted into training and evaluation sets and the model is evaluated based on evaluation metrics that include metrics like accuracy, precision, recall and f1-scores. For better visualization purposes, confusion matrix along with graphs between accuracy and loss with epochs were plotted. An accuracy of 95.06%, precision, recall, f1-scores of 0.9438, 0.9411, 0.9410 respectively and training time of 40 minutes concluded that transfer learning balances the performance and computational cost of the model unlike other deep learning models.

Keywords. Indian sign language (ISL), hand gesture recognition, image classification, MobilenetV2, transfer learning.

1 Introduction

Communication serves as a significant process in day-to-day life, while the major form of communication deals with speeches, verbal communication and written texts, there do exist disability specific communications and languages as well, out of which Sign language is one. Sign language is a vital mode of communication for the hearing-impaired community. It can be termed as a non-verbal mode of communication that uses visuals, hand gestures, body language and facial expressions to exchange information [2]. As per the details mentioned in the National Sample Survey (NSS) 76th round held in 2018, approximately 2.2% of the Indian population has a disability, with hearing impairment accounting for about 18.9% of these cases [20]. This translates to roughly 0.42% of the total population experiencing hearing disabilities, which highlights the importance of sign language in today's world. Apart from this, it has also been found that the non-verbal form of communication is more effective to be interpreted as compared to verbal communication. The

doi: 10.13053/CyS-29-3-5894

results highlighted and showcased the presence of a strong bond and interdependence among the quality, quantity and the method that the teachers used to involve non-verbal communication for educational purposes. Based on the results and findings of the review, it was revealed that the education process and students' academic growth were directly proportional to the nonverbal cues used by the teachers [5]. There are varying gesture languages across the world that include Indian Sign Language(ISL), ASL, Arabic Sign Language(ArSL), Italian Gesture Language, etc.

The differences extend beyond vocabulary to how signs are executed, incorporating non-manual markers and cultural references. Integrating technology can further bridge communication gaps for sign language users. Video conferencing solutions facilitate remote interpretation services, making it easier for specially abled employees to engage in meetings and conversations [22]. The vast diversity of the sign languages, along with their universality, is well debated and broadly analyzed by Evans & Levinson's (2009) example of sign language [7]. Sign languages form an alternative human linguistic system and vividly illustrate the rich diversity of language, along with the remarkable adaptability of the human mind and body in expressing it.

In recent years, sign language recognition has advanced through various methodologies. Traditional approaches utilized computer vision techniques like convex hull and contour-based methods to detect hand gestures. However, these methods often struggled with complex backgrounds and varying lighting conditions. The advancement of deep learning algorithms has significantly enhanced recognition accuracy. CNNs have been trained to learn spatial hierarchies of features, effectively capturing intricate patterns in hand gestures [19]. Long Short-Term Memory (LSTM) networks, a vanishing gradient alternative of recurrent neural networks, also serve as an enhanced technique for sign language detection, particularly for continuous sign detection. ResNet convolutional network has been used as the backbone model in LSTM-based approaches [11]. The past research has also implemented Xception and Inception combined with LSTM techniques that have yielded exceptional results and accuracy [3]. Another methodology, known as the Swin Transformer, a hierarchical vision transformer employing shifted windows, has been successfully utilized for sign language recognition tasks, offering notable advancements in both isolated and continuous gesture analysis. In a study focusing on Arabic Alphabet Sign Language, the Swin Transformer was fine-tuned on datasets like ArSL2018 and AASL, achieving impressive recognition accuracy [14]. Even though all these models have gained immense success, Transfer learning has become a crucial and influential approach in enhancing sign language recognition systems, particularly when dealing with limited datasets. By incorporating existent knowledge from models trained before hand on large datasets, transfer learning enables the development of effective and proficient gesture detectors without the necessity for extensive sign-specific data collection [21] [9]. Keeping in mind the significance of transfer learning, it has been used in this model.

The objective of this paper is to detect the hand gestures and signs used for communication through Indian Sign Language (ISL), thereby correctly labeling the signs to a class that represents a certain gesture, aiming to bridge the gap that exists in the interaction between normal and hearing-impaired people. Through this project, we contribute a robust and efficient training learning model for recognizing Indian Sign Language (ISL) gestures using transfer learning with MobileNetV2. This approach enhances model generalization through extensive image augmentation and careful fine-tuning of network layers. It provides a detailed evaluation framework, including per-class performance metrics and visualization tools for better interpretability. This work not only contributes towards showcasing the power of transfer learning in gesture recognition but also serves as a scalable foundation for assistive communication technologies, providing intellectuals with the opportunity to work on this technology in the future thereby, contributing towards the improvement of the scope of sign language. Besides this, the work is completely based on a newer idea of leveraging transfer learning with a pre-trained model that serves effective utilization of existing resources. It aims to reduce the gap between the hearing-disabled community and mainstream digital communication.

The dataset consists of thousands of ISL hand signs, classifying them into a total of 36 classes (26 alphabet + 10 digits) with each class representing a certain gesture or sign, that underwent preprocessing through techniques like grayscale conversion, normalization, and augmentation to enhance model accuracy before proceeding to model training and testing.

2 Literature Review

This section highlights the key research contributions that have shaped the current landscape of sign language recognition, shedding light on the major breakthroughs that have been achieved while also addressing the potential opportunities and challenges that researchers and developers may encounter in the future.

Antad et al. [3] implemented Convolutional Neural Networks(CNNs) along with other deep learning algorithms or methodologies firstly to recognise ISL and ASL thereby converting the gestures into written texts followed by conversion of the common text into different Indian regional languages. They obtained significant accuracy rates of 99.72% and 99.90% for recognising color diversified images and greyscale images respectively. Mohammed et al. [15], proposed a study to overcome limitations of Yemeni Sign language(YSL) by extending a new dataset and increasing the efficiency of transfer learning apporaches.The authors adopted deep learning models as such AlexNet , ResNet152V2, Swin Transformer, InceptionV3 and Xception. The dataset was collected with 24,245 sign images over a diverse range of 32 class labels.They encountered rich success with highest validation and testing accuracies of 98.80% and 99.00%.

Wadhawan et al. [24], dealt with a study that focused on resilient modelling of non-continuos signs within the domain of sign language translation through deep learning techniques based CNNs. Performance was assessed using various optimizers, revealing that the proposed method attained the best training accuracies of 99.72% for colored images and 99.90% for grayscale images. Utilizing the SGD optimizer, the model obtained training and validation accuracies of 99.72% and 98.56%, respectively.

In a research work conducted by Shaba et al. [23], Various deep learning techniques including CNNs, VGG16, and ResNet, were employed, and evaluated performances of the models were compared using accuracy, precision, recall,and F1-Score metrics. CNN obtained a remarkable accuracy of 99.98% with 10 rounds of training(epochs) and a batch size of 64. The validation accuracy of the model is 96%. VGG16 achieved a second remarkable accuracy of 98.29% and correctly predicted 673 images, while ResNet achieved the lowest accuracy among all the models with 96.43% and correctly predicted 670 images.

Al-Hammadi et al. [1], proposed an effective deep convolutional neural network, leveraging transfer learning approaches that address the limitation of having a small labeled hand gestures and signs dataset. The model was tested and evaluated on three color video-based gesture datasets, comprising 40, 23, and 10 gesture classes. In the signer-dependent mode, the method achieved recognition accuracy of 98.12%, 100%, and 76.67% on the respective datasets.

As a new approach for recognising Arabic Sign language(ArSL), Farouk et al. [8], implemented Media-Pipe, a robust framework for real-time hand and pose detection. For static sign language recognition, TensorFlow Lite was integrated, which achieved an overwhelming accuracy of 99.1% utilizing a CNN model and 99.5% with a SVM model.

For dynamic sign language, they implemented a LSTM model within TensorFlow, that obtained an accuracy of 93%. M. A. A. Mosleh and A. H. Gumaei [18], proposed a prototype that leveraged CNN based deep learning approaches and fuzzy string matching utilities for efficient sign recognition and text conversion. The proposed system obtained remarkable translation accuracy rates for various CNN architectures, with ResNet152 model achieving 98.78%, MobileNet V1 achieving 97.94%, GoogleNet attaining 98.36%, VGG16 scoring 90.46%, and DenseNet161 obtaining 98.34% from the basis of experimental results.

Many individuals are dependent on sign and gesture languages as their sole means of interaction, but accessibility remains a challenge in various aspects of daily life, such as education, employment, and public services. This motivates the requirement of an efficient sign language recognition system developing which, we can facilitate interaction between sign language users and non-signers. Despite advancements, many existing models are unable to meet expected performance with real-time processing, scalability, and generalization across diverse sign languages.

Motivated by the importance of real-time, efficient sign language translation, deep learning as well as transfer learning solutions have been explored in this study. MobileNetV2, known for its lightweight architecture, enables fast and accurate classification while maintaining computational efficiency. Transfer learning further aids model performance by leveraging pre-trained knowledge, improving training time and data dependency.

By integrating these features, the goal is to develop an accessible and scalable sign language detection system, bridging communication gaps and promoting inclusivity.

3 Proposed Methodology

The model implements a MobileNetV2 architecture which directly learns from data and recognizes patterns in the images according to their classifications. The first step involves implementing the dataset using images from a dataset source to train the ISL model.

Afterwards the data is preprocessed and augmented, i.e., augmenting the dataset with alterations like rotation, cropping, scaling, etc. For this model, the dataset is split into 80:20 ratio for training and evaluation respectively.

The last phase includes testing the trained model with new and unseen images and determining the accuracy based on its detected hand gestures and translation. A schematic representation of the described framework is illustrated in Figure 1.

3.1 Dataset Description

The dataset was developed using hand gestures and signs conveyed on the Indian Sign Language Research and Training Centre (ISLRTC) website, with 1000 images being included per character for ISL (https://www.kaggle.com/datasets/atharvadumbre/indian-sign-language-islrtc-referred).

A total of 36000 images were included in the dataset, which were divided into 36(26 alphabets + 10 digits) class labels with each class label containing 1000 gesture images and representing a different class of hand gesture or sign used in ISL. All the images in the dataset have been standardized to a size 250×250 pixels. This uniformity validates consistency in data presentation and facilitates ease of preprocessing and model development. Each of the labelled classes consisting of 1000 sample images or gestures, were proceeded further for preprocessing and augmentation before the model training and evaluation begins.

The bar graph representing the frequency distribution of the sample images over the labeled classes maintain a strict uniformity and can be visualized through a Bar graph as shown in Figure 2. The x-axis represents the class indices (0-36) for labeled classes 0,1,. . . ,9, and A,B,. . . , Z and the y-axis indicates the number of images each class label consists of .

3.2 Data Preprocessing and Augmentation

The preprocessing stage in this deep learning pipeline was responsible for preparing the dataset before training. This involved data augmentation, normalization, resizing, and data loading for model training. Before training the model, the input images were converted to gray scale to reduce complexity and improve feature extraction.

Since color information is not essential for recognizing hand signs, gray scaling helps the model focus on shape and texture. The images were resized to 224×224 pixels, normalized to [0, 1], and structured into batches for efficient training. This transformation ensured the dataset was optimized for learning while maintaining essential sign language features. The batch size was set to

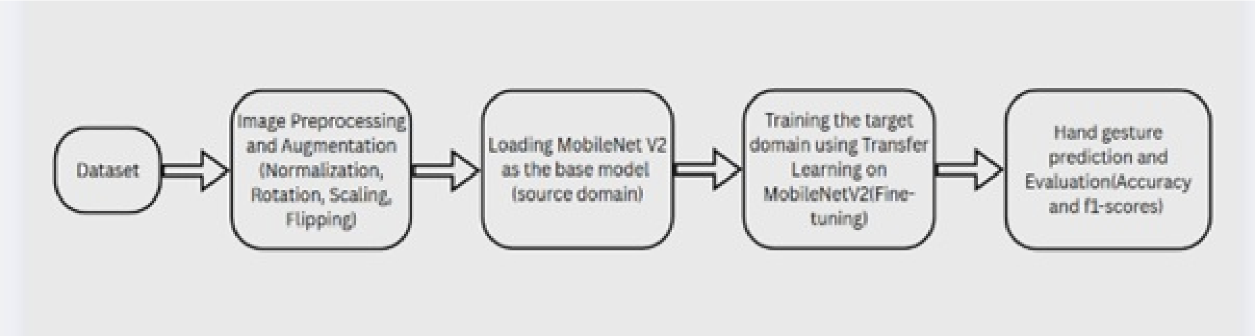

Dataset
Image Preprocessing and Augmentation (Normalization, Rotation, Scaling, Flipping)
Loading MobileNet V2 as the base model (source domain)
Training the target domain using Transfer Learning on MobileNetV2(Fine-tuning)
Hand gesture prediction and Evaluation(Accuracy and f1-scores)

Fig. 1. Model framework for sign language detection

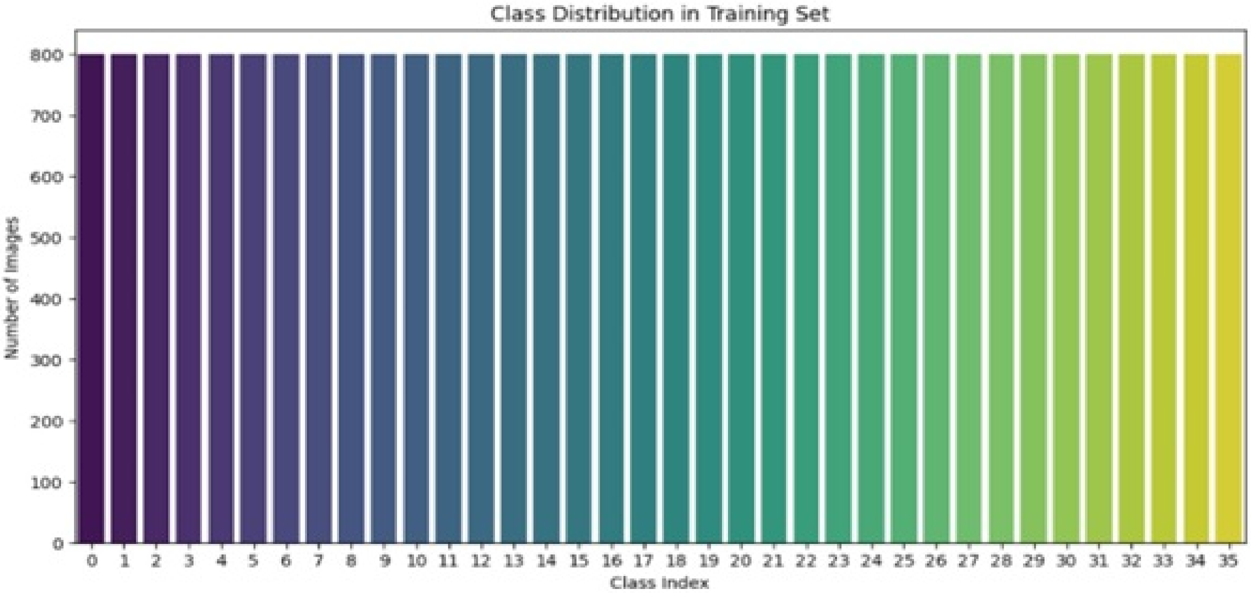


Fig. 2. Frequency distribution Of images over class indices

2, meaning 32 images were processed per batch. The dataset was augmented with transformations like rotation, scaling, and flipping to increase diversity. The data augmentation operations involved rescaling where the pixel values were converted from [0,255] to [0,1] for better numerical stability.

Then Shear transformation was implemented in which a slight distortion was applied to images followed by zoom augmentation where the images were randomly zoomed. At last, the images underwent flipping in order to improve model generalization.

3.3 Model Development

Building an effective sign language recognition system necessitates choosing a suitable model that can accurately interpret hand gestures.

A range of machine learning approaches as well as deep learning techniques can be considered for this objective, each offering its own set of strengths and drawbacks. Conventional machine learning models like Random Forest classification, SVMs and Logistic Regression have been extensively utilized in image classification problems. However, their effectiveness in sign language recognition is limited due to their reliance on handcrafted feature extraction. So the algorithm or technique that this model implements is based on transfer learning combined with a pre-trained model CNN model, i.e., MobileNet V2. The CNN model is fine-tuned with different transfer learning techniques to achieve a higher rate of accuracy.

3.3.1 MobileNetV2

The MobileNetV2 model can be defined as a type of CNN, optimized for efficient image classification tasks. It is a lightweight deep learning based architecture and framework designed for efficient feature extraction, particularly on mobile and embedded devices [10]. MobileNetV2 as a Pre-trained CNN Feature Extractor has been used in this model, initially loading with ImageNet weights. The features are then processed by a bCNN classification head. The architecture of MobileNetV2 consists of Depth wise separable Convolution, Inverted Residuals and Bottleneck Design. The MobileNetV2 architecture is represented in Figure 3 with the following components:

i. Input: The input is a color image(3 channels) with dimensions of 128×128 that has underwent preprocessing including Normalization, Resizing, etc.

ii. 3×3 Convolution and ReLU: It indicates the initial convolution layer with ReLU activation and outputting feature maps. The output size after convolution changes to 64×64 with n = 32 feature maps.

iii. 2×2 Max Pool: It down samples the feature map by taking the max over 2×2 patches, reducing spatial dimensions.

iv. Final feature map: Size of the output is $4 \times 4 \times 1280$ with 1280 being the final output channel count. The flatten layer then converts the 3D feature map into a 1D vector of size 20480, which feeds into the dense layer.

v. Classifier: It is the fully connected layer consisting of a Dense layer and Output classes.

3.3.2 Transfer Learning

Transfer learning has become a highly effective strategy within the realm of deep learning methods that works on the principle of reusing knowledge obtained from an existing task, referred to as the source or base domain, and applying it to a non-identical but related task referred to as the task domain. It is a popular technique that aids MobileNetV2 thereby extending the use of models trained beforehand to large-scale datasets.

The use of transfer learning can be justified by its features matching the requirements of this model as the dataset is significantly large with a total of 36,000 sample images that would normally be time-consuming in training the model. Besides, the labeled data is not extensive which can be compromised by incorporating pre-learned characteristics and models from the base domain thereby highlighting the significance of transfer learning. In a standard transfer learning setup, a model is first trained through a large-scale dataset from a base domain, enabling it to learn broad and transferable features. This initial training significantly equips the model with a strong foundation of general representations.

The trained model is then adjusted by means of fine-tuning techniques to perform well on a small-scale, domain-scoped dataset inherited from the target task. By building on the knowledge gained during pre-training, the model enhances its performance in the target domain.

3.4 Model Training

Once the dataset is preprocessed and augmented, it is splitted into a ratio of 80:20 validation split, i.e, 80% of the images are put under train_generator for training and the remaining 20% are put under test_generator for validation, maintaining a separable boundary between both sets to ensure the proper evaluation of the model. The train generator loads images into batches of 32 and uses categorical labels since there are 36 classes.

Then the pre-trained model (MobileNetV2) is loaded as the base model. Its weight is set as “imagenet”, thereby using pretrained weights from ImageNet. To remove the original classification head, the “include_top” parameter of the model

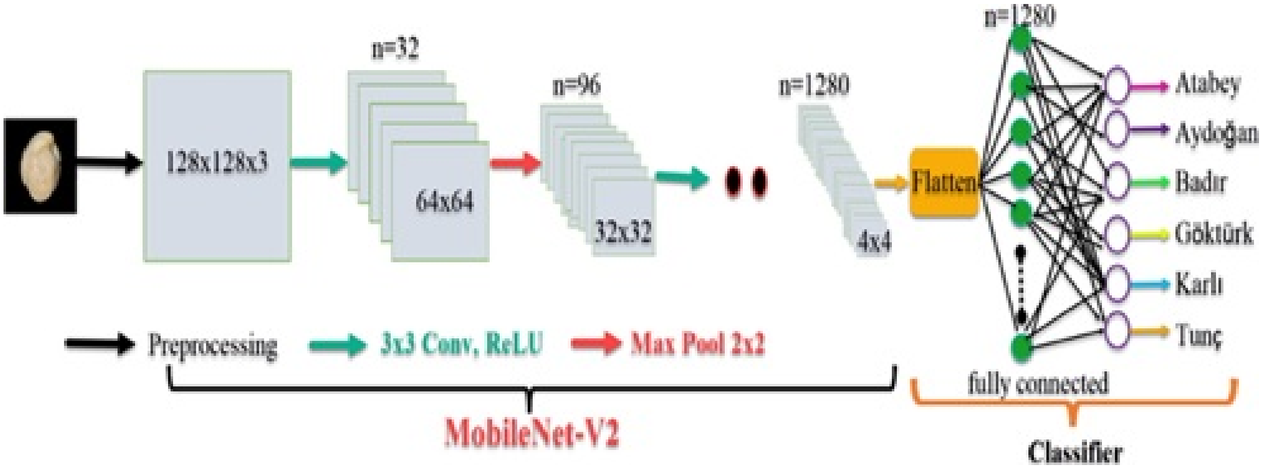


Fig. 3. MobileNetV2 architecture

is set to false. The base layers and initial layers were then frozen, preventing weight updates in pre-trained models and ensuring only newly added layers are trained. However, the last few layers (30 layers) were unfrozen for fine-tuning. Since the original classification head was removed, new classifier with 128 Neurons was added and a softmax layer for multi-class classification of the images.

Then the model undergoes compilation with optimizer set as “adam” and loss function set as “categorical cross-entropy”. The Adam (Adaptive Moment Estimation) optimizer [16] automatically adjusts the learning rate for each parameter, combining 2 optimisation techniques, i.e., Momentum (from SGD) and RMSprop, that use past gradients to accelerate learning. Categorical cross-entropy [6] is used as a loss function here, which calculates how different the predicted probability distribution is from the actual probability prediction. After the model is compiled, it proceeds to the training stage with the train generator.

The train generator loads batches of images from the dataset, and one-hot encoded labels are automatically assigned. In the Forward Propagation, the batch is passed through MobileNetV2. Each layer processes the images using Convolution, Pooling and Activation functions. The last dense layer outputs a probability for each class (softmax activation). Next, the predicted output or class label is compared with actual labels using categorical cross-entropy. So the error is calculated and propagated backwards through the network in the Back Propagation and Weight Update [17] stage. Adam optimizer adjusts the model’s weights using gradient descent, thereby helping the model make better predictions in the next epoch. A total of 8-10 epochs has been designated for the model. At the end of each epoch, the model undergoes the evaluation stage, where the model is tested on test_generator or validation_generator. The validation loss & accuracy, and all other required metrics are calculated to check how well the model is generalizing and predicting the class labels.

3.5 Model Evaluation

The performance of the trained model is evaluated using a 20% validation or test split. Various metrics are employed for evaluation, including accuracy, recall, precision, loss, and F1-score. At the conclusion of each round of training, the model is evaluated on basis of the test generator dataset to measure how closely its predictions align with the true class labels of the images. Accuracy serves as the major evaluating factor which is being computed for both the performance of the model on the train generator as well as the validation or test generator. The training accuracy being high and test accuracy being low will highlight if there

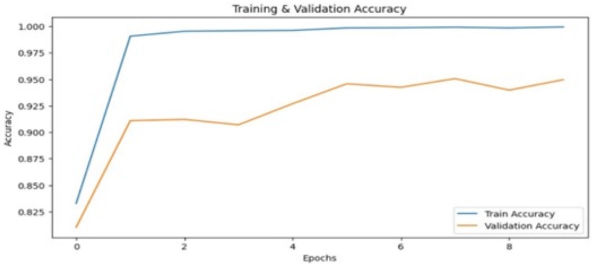


Fig. 4. Training and validation accuracy of MobileNet V2+Transfer learning model against epochs

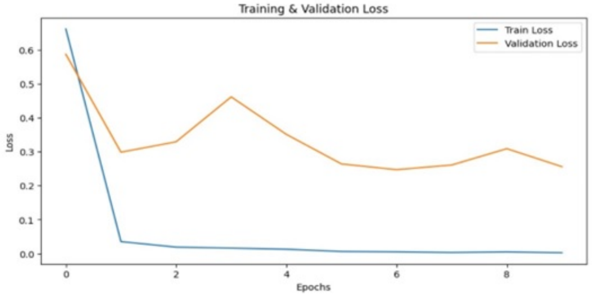


Fig. 5. Training and validation loss of MobileNet V2+Transfer learning model against epochs

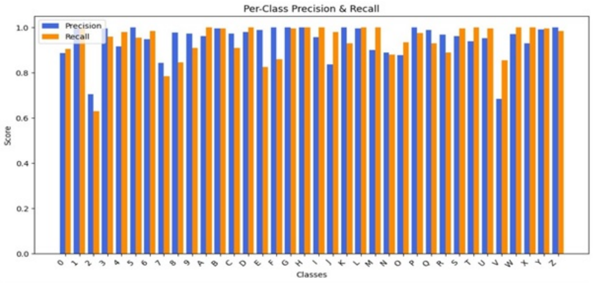


Fig. 6. Precision and Recall against each class index

is the case of Overfitting and both of them being low will highlight the case of Underfitting. Apart from all these evaluation metrics, the confusion matrix (Misclassification Analysis) is used to help vizualise how many instances of each classes were correctly identified or predicted.

4 Result and Discussion

4.1 Experimental Setup

The model is developed and executed in a Google Colab environment, leveraging its built-in GPU support for efficient deep learning training.

The dataset is structured in folders under the /content/ISL_Dataset_filtered directory, where each subfolder represents a class of certain hand gestures. The training pipeline uses TensorFlow and Keras, with image augmentation powered by ImageDataGenerator for robustness. The MobileNetV2 model is used as the base, fine-tuned for sign language classification. Essential libraries such as NumPy, Pandas, Matplotlib, and Seaborn are used for data manipulation and visualization. GPU acceleration, combined with callback mechanisms like ReduceLR, OnPlateau and EarlyStopping, ensures optimal model training and generalization. GPU support helps the model to be trained within a minimal time.

Using different Python libraries available in Google Colab environment, the metrics for the model were evaluated, and the graphs were plotted using pyplot for better visualization. The designed MobileNetV2 Transfer learning model attained a highest accuracy of 95.06% on the validation set, while it reached consistently an accuracy of more than 99.8% on training sets corresponding to each epoch. The variance of training and validation accuracies with respect to the epochs can be visualized through the graph, as shown in Figure 4.

Next, the loss metric for each epoch is computed through the technique used for multi-class classification problems, known as categorical cross-entropy. The training and validation losses were plotted against epochs as illustrated in the graph shown in Figure 5. The graph clearly suggests that the model shows no sign of Overfitting(Low training loss but high validation loss) or Underfitting (Both high training and validation loss), thereby indicating that training and validation loss decrease together, enhancing the stability of the model.

The Precision, Recall and f1-scores were also computed to analyse the working of the model and how well the predictions are as compared to the actual labels. The Precision and recalls were plotted against each class index as highlighted in Figure 6.

A classification report was generated to highlight the metrics for each of the 36 class labels, including the digits 0. . . 9 and the alphabet A. . . Z for the validation sample of 7200 images, which is shown in Table 1. The overall metrics evaluated after the complete training of the model are shown in Table 2:

The evaluation and computation of all these matrices can be visualized by the means of a table, known as the Confusion matrix which is used to compare actual class labels with the predicted class labels of a classification model. It is a square matrix, with its dimensions equal to the class labels or indices so the confusion matrix in this model is a 36×36 matrix where the numbers present on the main diagonal of the matrix denote the number of correct class labels predicted by the model and the numbers present off the main diagonal denote the incorrect predictions made by the model. The confusion matrix is shown Figure 7.

5 Comparison

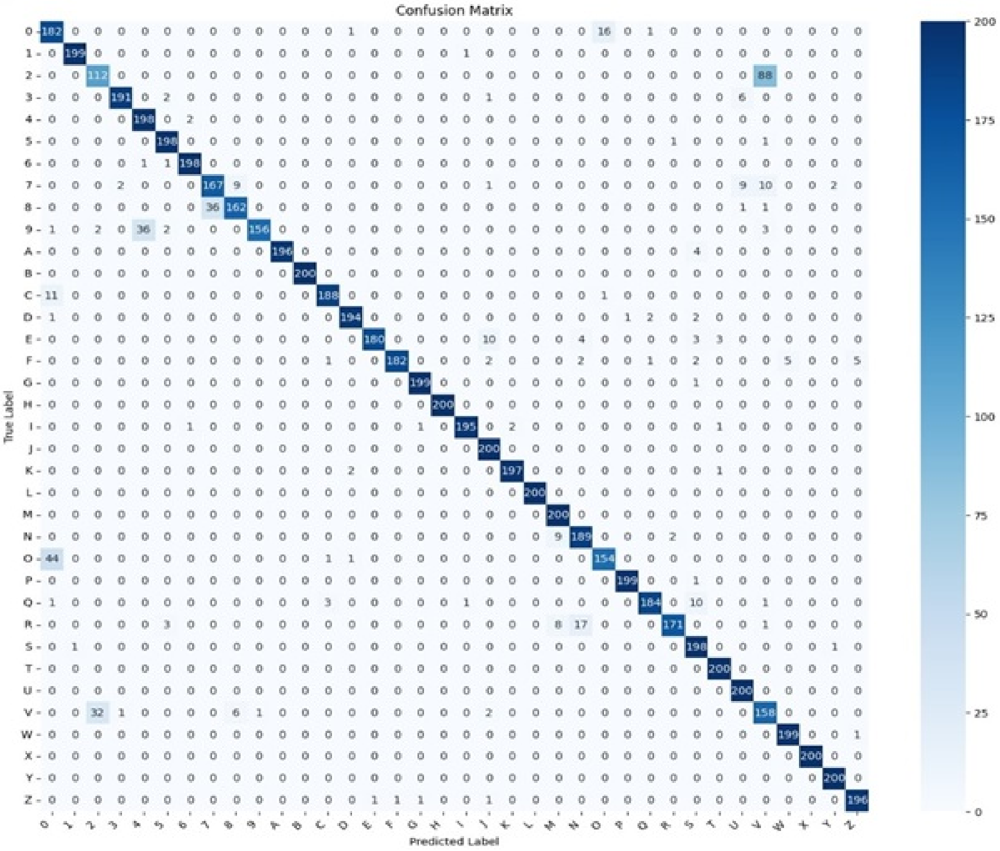
Various models have been explored for their efficiency under the scope of sign language translation. A study on the ArASL2018 dataset reported that MobileNetV2 obtained a validation accuracy of 99.48% with a training time of 26.52 minutes, along with 98.91%, 98.56%, and 98.69% precision, recall and f1, respectively, while ResNet50 attained 99.30% accuracy in 60.94 minutes with precision, recall and f1-score of 99.31%, 99.45% and 98.98%.

Transformer-based models like Microsoft's Swin reached 99.60% accuracy but required 580.50 minutes for training, indicating a trade-off between accuracy and computational efficiency [4]. Another research introduced a hybrid framework that integrated MobileNetV3, multi-head self-attention mechanisms, and LightGBM. This approach achieved a validation accuracy of 98.42%, with precision, recall and F1-scores of 98.19%, 98.81%, and 98.15% respectively, completing training in 35.30 minutes [13].

Additionally, a stacked encoded deep learning framework utilizing EfficientNetB3 with encoder

Table 1. Experimental metrics for individual classes

Class label	Precision	Recall	F1-score	Support
0	0.89	0.91	0.90	200
1	0.99	0.99	0.99	200
2	0.70	0.63	0.66	200
3	0.99	0.96	0.98	200
4	0.92	0.98	0.95	200
5	1.00	0.95	0.98	200
6	0.95	0.98	0.97	200
7	0.84	0.79	0.81	200
8	0.98	0.84	0.81	200
9	0.97	0.91	0.94	200
A	0.96	1.00	0.98	200
B	0.99	0.99	0.99	200
C	0.97	0.91	0.94	200
D	0.98	1.00	0.99	200
E	0.97	0.82	0.90	200
F	1.00	0.86	0.92	200
G	1.00	0.99	1.00	200
H	1.00	1.00	1.00	200
I	0.96	1.00	0.98	200
J	0.84	0.98	0.90	200
K	1.00	0.93	0.96	200
L	1.00	1.00	1.00	200
M	0.90	1.00	0.95	200
N	0.89	0.88	0.88	200
O	0.88	0.94	0.91	200
P	1.00	0.97	0.99	200
Q	0.99	0.93	0.96	200
R	0.97	0.89	0.93	200
S	0.96	0.99	0.98	200
T	0.94	1.00	0.97	200
U	0.95	0.99	0.97	200
V	0.68	0.85	0.76	200
W	0.97	1.00	0.99	200
X	0.93	1.00	0.96	200
Y	0.99	0.99	0.99	200
Z	1.00	0.98	0.99	200



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	182	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	16	0	1	0	0	0	0	0	0	0	0	0
1	0	199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	88	0	0	0	0
3	0	0	0	191	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
4	0	0	0	0	198	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	198	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
6	0	0	0	0	1	1	198	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	2	0	0	0	167	9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	9	10	0	0	2	0
8	0	0	0	0	0	0	0	36	162	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
9	1	0	2	0	36	2	0	0	0	156	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	196	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	11	0	0	0	0	0	0	0	0	0	0	0	188	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
D	1	0	0	0	0	0	0	0	0	0	0	0	0	194	0	0	0	0	0	0	0	0	0	0	0	1	2	0	2	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	0	0	0	10	0	0	0	4	0	0	0	0	3	3	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	182	0	0	0	2	0	0	0	2	0	0	1	0	2	0	0	0	5	0	0	5
G	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	195	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	197	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	189	0	0	0	2	0	0	0	0	0	0	0	0
O	44	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	1	0	0	0	0	0	0	0
Q	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1	0	0	0	0	0	0	0	184	0	10	0	0	1	0	0	0	0
R	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	17	0	0	0	171	0	0	0	1	0	0	0	0
S	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	198	0	0	0	0	0	1	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0	0	0	0
V	0	0	32	1	0	0	0	0	6	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	158	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	0	0	1
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	200	0
Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	196

Fig. 7. Confusion matrix

Table 2. Overall Evaluation Metrics

Metrics	Values
Accuracy	95.06%
Precision	0.9438
Recall	0.9411
F1-score	0.9410

and decoder networks reported an accuracy of 99.26% [12]. These findings suggest that while MobileNetV2 offers a balance between accuracy and efficiency, integrating advanced architectures like Swin transformers or hybrid models can enhance performance, albeit with increased computational demands. This MobileNetV2-based sign language recognition model balances accuracy and efficiency, achieving 95.06% accuracy with a significantly lower computational cost compared to ResNet and Swin Transformer, which demand higher training time.

Unlike transformer-based models that require extensive data and resources, this model maintains strong precision (0.9438) and recall (0.9411) while being lightweight and suitable for real-time applications.

Additionally, CNN-based deep learning models often struggle with overfitting, whereas this transfer learning approach ensures generalization across varied sign language augmented gestures. The model's efficient architecture also makes it deployable on edge devices, unlike computationally expensive alternatives. A comparison of the metrics for the different models is illustrated in Table 3.

Table 3. Comparision of different models

Model	MobileNet V2 + TL	ResNet	Swin Transformer	MobileNet V3
Accuracy	99.48	99.30	99.60	98.42
Precision	98.91	99.31	96.78	98.19
Recall	98.56	99.45	97.90	98.91
F1-score	98.69	98.98	98.75	98.15
Training Time(min)	26.52	60.94	580.50	35.30

6 Conclusion

The surveys and findings conclude that sign language lately has been a major center of researches and works thereby justifying the different technological advances in the domain of sign language detection and recognition. Despite all these advancements and impressive accuracy attained by the past models and journals related to this domain it has still been a challenge to balance the efficiency with the resources and time required to be invested. This work exactly aimed to attain that balance thereby not compromising the demands and computational time of the model with its performance.

The work successfully demonstrates the use of transfer learning to recognize Indian Sign Language (ISL) gestures. By leveraging MobileNetV2 and effective image augmentation, the model achieves reliable classification performance. Visualizations such as accuracy plots, confusion matrix, and per-class precision/recall enhance interpretability. The system performs well even with class imbalances, indicating robustness. Overall, it represents a significant step toward AI-powered communication tools for the hearing and speech impaired. The model attained a remarkable accuracy of 95.06% along with precision, recall, f1-scores of 0.9438, 0.9411, and 0.9410 with a training time of nearly 40 minutes that overcomes the high training or computation time required by other models.

While the model shows strong performance, futuristic advancements could include extending the scope of the dataset to include more sign variations and real-time gestures. Integrating live webcam input with real-time prediction and gesture sequence detection using LSTMs or transformers can enhance practical usability.

Additionally, deploying the model via mobile or web applications could improve accessibility for the broader community. Domain adaptation for different lighting conditions or hand orientations would further boost generalizability.

Acknowledgments

We would like to thank all the anonymous reviewers for their constructive suggestions to make this work effective for the target audience.

References

1. **Al-Hammadi, M., et al. (2020).** Hand gesture recognition for sign language using 3dcnn. IEEE Access, Vol. 8, pp. 153270–153284. DOI: 10.1109/ACCESS.2020.3018480.

2. **Alaghband, M., Maghroor, H. R., Garibay, I. (2023).** A survey on sign language literature. Machine Learning with Applications, Vol. 14, pp. 100504. DOI: 10.1016/j.mlwa.2023.100504.

3. **Antad, S. M., Chakrabarty, S., Bhat, S., Bisen, S., Jain, S. (2024).** Sign language translation across multiple languages. 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), IEEE, pp. 741–746. DOI: 10.1109/ESIC60604.2024.10481626.

4. **Balat, M., Awaad, R., Adel, H., Zaky, A. B., Aly, S. A. (2024).** Advanced arabic alphabet sign language recognition using transfer learning and transformer models. 2024 International Conference on Computer and Applications (ICCA), IEEE, pp. 1–6.

5. **Bambaeeroo, F., Shokrpour, N. (2017).** The impact of the teachers' non-verbal communication on success in teaching. Journal of Advances in Medical Education & Professionalism, Vol. 5, No. 2, pp. 51–59.

6. **Chan, J., Papaioannou, I., Straub, D. (2024).** Bayesian improved cross entropy method with categorical mixture models for network reliability assessment. Reliability Engineering & System Safety, Vol. 252, pp. 110432. DOI: 10.1016/j.ress.2024.110432.

7. **Evans, N., Levinson, S. C. (2009).** The myth of language universals: Language diversity and its importance for cognitive science. Behavioral and Brain Sciences, Vol. 32, pp. 429–492. DOI: 10.1017/S0140525X0999094X.

8. **Farouk, A. M., Zenhom, A. M., Abdelaleem, E. M., Fadel, S. A., Elsayed, K. K. A., Mohammed, M. S., Hassan, R. H., Shedeed, H. A. (2024).** A new approach for arabic sign language recognition (arslr). 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES), IEEE.

9. **Gholizade, M., Soltanizadeh, H., Rahmanimanesh, M. (2025).** A review of recent advances and strategies in transfer learning. International Journal of System Assurance Engineering and Management. DOI: 10.1007/s13198-025-02119-3.

10. **Gulzar, Y. (2023).** Fruit image classification model based on mobilenetv2 with deep transfer learning technique. Sustainability, Vol. 15, No. 1906. DOI: 10.3390/su15031906.

11. **Huang, J., Chouvatut, V. (2024).** Video-based sign language recognition via resnet and lstm network. Journal of Imaging, Vol. 10, No. 6, pp. 149. DOI: 10.3390/jimaging10060149.

12. **Islam, M., Aloraini, M., Aladhadh, S., Habib, S., Khan, A., Alabdulatif, A., Alanazi, T. M. (2023).** Toward a vision-based intelligent system: A stacked encoded deep learning framework for sign language recognition. Sensors, Vol. 23, No. 22, pp. 9068.

13. **Kumar, H., Sachan, R., Tiwari, M., Katiyar, A. K., Awasthi, N., Mamoria, P. (2025).** Hybrid sign language recognition framework leveraging mobilenetv3, multi-head self-attention and lightgbm. Journal of Electronics, Electromedical Engineering & Medical Informatics, Vol. 7, No. 2, pp. 318–329.

14. **Kumar, Y., Huang, K., Lin, C.-C., Watson, A., Li, J. J., Morreale, P., Delgado, J. (2024).** Applying swin architecture to diverse sign language datasets. Electronics, Vol. 13, No. 1509. DOI: 10.3390/electronics13081509.

15. **Mohammed, A. A. A., Esmail, E., Mosleh, M. A. A., Mohammed, R. A. A., Almuhaya, B. (2024).** Development and evaluation of pre-trained deep learning models for efficient arabic sign language recognition. 2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA), IEEE.

16. **Mohan, A., Mohan, D., Vats, S., Sharma, V., Kukreja, V. (2024).** Classification of sign language gestures using cnn with adam optimizer. 2024 2nd International Conference on Disruptive Technologies (ICDT), IEEE, pp. 430–433. DOI: 10.1109/ICDT61202.2024.10489158.

17. **Momeni, A., et al. (2023).** Backpropagation-free training of deep physical neural networks. Science, Vol. 382, pp. 1297–1303. DOI: 10.1126/science.adi8474.

18. **Mosleh, M. A. A., Gumaei, A. H. (2024).** An efficient bidirectional android translation prototype for yemeni sign language using fuzzy logic and cnn transfer learning models. IEEE Access, Vol. 12, pp. 191030–191045. DOI: 10.1109/ACCESS.2024.3512455.

19. **Najib, F. M. (2025).** Sign language interpretation using machine learning and artificial intelligence. Neural Computing and Applications, Vol. 37, pp. 841–857. DOI: 10.1007/s00521-024-10395-9.

20. **National Statistical Office, Ministry of Statistics and Programme Implementation, Government of India (2018).** Persons with disabilities in india: Nss 76th round (july – december 2018). Technical report.

21. **Ridwan, A. E. M., Chowdhury, M. I., Mary, M. M., Abir, M. T. C. (2024).** Deep neural network-based sign language recognition: A comprehensive approach using transfer learning with explainability.

22. **Sandler, W. (2010).** The uniformity and diversity of language: Evidence from sign language. Lingua, Vol. 120, No. 12, pp. 2727–2732. DOI: 10.1016/j.lingua.2010.03.015.

23. **Shaba, S. A., Shroddha, S. I., Hossain, M. J., Monir, M. F. (2024).** Neuralgesture communication: Translating one sign language to another using deep learning model and gtts. 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), IEEE, pp. 1–5. DOI: 10.1109/VTC2024-Spring62846.2024.10683444.

24. **Wadhawan, A., Kumar, P. (2020).** Deep learning-based sign language recognition system for static signs. Neural Computing and Applications, Vol. 32, pp. 7957–7968. DOI: 10.1007/s00521-019-04575-4.

Article received on 29/04/2025; accepted on 27/07/2025.
**Corresponding author is Sanjit Kumar Dash.*