*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

# Enhancing Sign Language Recognition through Fusion of CNN Models

**Ajay M. Pol[*1,2], Shrinivas A. Patil [3]**

[1]Research Scholar, Shivaji University, Kolhapur, India.
[2]Assistant Professor, Department of Electronics and Telecommunication, KIT's College of Engineering (Autonomous), Kolhapur, India.
[3]Professor and Head, Department of Electronics and Telecommunication, DKTE Societies' Textile and Engineering Institute (Autonomous), Ichalkaranji, Kolhapur, India.
*Corresponding Author (Email): pol.ajay@kitcoek.in
ORCID ID: 0000-0002-3014-1395

**Abstract**— This study introduces a pioneering hybrid model designed for the recognition of sign language, with a specific focus on American Sign Language (ASL) and Indian Sign Language (ISL). Departing from traditional machine learning methods, the model ingeniously blends hand-crafted techniques with deep learning approaches to surmount inherent limitations. Notably, the hybrid model achieves an exceptional accuracy rate of 96% for ASL and 97% for ISL, surpassing the typical 90-93% accuracy rates of previous models. This breakthrough underscores the efficacy of combining predefined features and rules with neural networks. What sets this hybrid model apart is its versatility in recognizing both ASL and ISL signs, addressing the global variations in sign languages. The elevated accuracy levels make it a practical and accessible tool for the hearing-impaired community. This has significant implications for real-world applications, particularly in education, healthcare, and various contexts where improved communication between hearing-impaired individuals and others is paramount. The study represents a noteworthy stride in sign language recognition, presenting a hybrid model that excels in accurately identifying ASL and ISL signs, thereby contributing to the advancement of communication and inclusivity.

**Keywords**- Sign language recognition, Hybrid model, American Sign Language (ASL), Indian Sign Language (ISL), Deep learning.

## I. INTRODUCTION

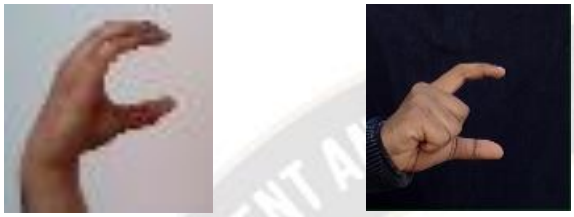
Sign Language is an innate language utilized by deaf individuals or having hearing loss to communicate amongst themselves and with hearing individuals who have acquired proficiency in the language [1]. The recognition of Sign Language has become increasingly important in recent years due to the growing population of deaf [2] and hard-of-hearing individuals and the need to create more inclusive and accessible environments. The main objective behind sign language recognition is to translate the meaning of it into written or spoken language form or the one which can be understood by computers. Multiple methodologies exist for Sign Language recognition, encompassing computer vision-based techniques, data-driven strategies, and deep learning-based methodologies [3]. Sign Language recognition is useful for hearing impaired people when applied in finger-spelling, isolated signs, continuous recognition, and translation [4]. Furthermore, Sign Language recognition can facilitate effective communication not only between deaf and hearing individuals but also between individuals who employ different Sign Languages.

Although significant advancements have been achieved in the field of Sign Language recognition, numerous challenges persist that need to be addressed [5]. These challenges include the variability of Sign Language, the lack of large-scale datasets, and the need for more robust and accurate recognition algorithms. There is difference in representing the numbers and alphabets in American Sign Language (ASL) [6] compared to Indian Sign Language (ISL) [7]. Firstly, ASL and ISL use different signs for numbers and alphabets. ASL uses a one-handed sign language alphabet, where each letter is represented by a unique handshape, while ISL uses a two-handed alphabet where each letter is formed using a combination of handshapes and movements. Secondly, the number signs in ASL and ISL differ in their form and structure. In ASL, the numbers are represented using handshapes that are specific to each number, while in ISL, the numbers are represented using a combination of handshapes and movements that represent the value of the number. Another key difference is the directionality of signs in the two languages. In ASL, most signs are made in front of the body, while in ISL, many signs are made to the side or behind the body. Finally, the grammar and sentence structure of ASL and ISL also differ. The structure of ASL is characterized by a subject-verb-object (SVO) arrangement, whereas ISL exhibits a subject-object-verb (SOV) structure. The example sign language images for ASL and ISL are shown in Figure 1.

Overall, while both ASL and ISL are sign languages, they have unique differences in their approach to numbers, alphabets, and grammar. Understanding these differences is essential for effective communication between individuals who use these languages. To tackle these obstacles, researchers have

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

introduced diverse models for sign language recognition. These encompass Convolutional Neural Networks (CNNs) like ResNet-50 and Inception V3, as well as attention-based models that concentrate on specific segments of the input sequence. In this context, a new hybrid model has been developed that combines a customized attention-based model with ResNet-50 and Inception V3. This model aims to leverage the strengths of each component to improve sign language recognition accuracy.



(a) ASL Alphabet 'C' (b) ISL Alphabet 'C'

Figure 1: Sign Language Example for ASL and ISL

The customized attention-based model focuses on capturing the temporal dependencies of sign language gestures, while ResNet-50 and Inception V3 are used to extract high-level features from the image and video data. These models are combined for a more comprehensive analysis of the input data, leading to improvement in recognition accuracy. The main contribution points of the proposed hybrid sign language recognition model that combines a customized attention-based model, ResNet-50, and Inception V3 include:

- Improved recognition accuracy: By leveraging the strengths of each component, the proposed model achieves better recognition accuracy compared to traditional CNN-based models.
- Robustness to noise: The attention-based model allows the network to adaptively focus on parts of the feature vectors, improving its ability to recognize sign language gestures even in the presence of background noise and other distractions.
- Generalization: The fusion of transfer learning allows the model to generalize to new and unseen sign language gestures and users, making it more versatile and adaptable to different applications and settings.
- Accessibility: The main interaction problem due to communication gap can be filled with the proposed model with accurate recognition of sign language.

In summary, the hybrid sign language recognition model presents a significant stride in the advancement of sign language recognition systems. Its potential impact on enhancing communication and accessibility for the deaf and hearing-impaired communities is substantial.

## II. METHODOLOGY

The methodology for improving sign language recognition performance involves three key steps: dataset preparation, attention model development, and fusion of transfer learning. Firstly, a comprehensive dataset is prepared from standard public resources, which includes a large number of sign language gesture examples and their corresponding labels. This dataset is used to train and validate the model. Next, an attention-based model is developed that is capable of capturing the temporal dependencies of sign language gestures. This approach enables the neural network to concentrate on particular segments of the input sequence, thereby enhancing its capacity to discern subtle distinctions among comparable gestures. Finally, transfer learning is used to further improve the model's performance. Specifically, ResNet-50 and Inception V3 models are used in transfer learning approach to extract high-level features from the image and video data. The concatenation of these features are then made with the attention-based model to create a hybrid model that can better analyze the input data.

Through the fusion of these models, the resultant model can effectively harness the individual strengths of each component, leading to enhanced accuracy in sign language recognition. This approach signifies a significant breakthrough in the evolution of sign language recognition systems, holding immense potential in elevating communication and accessibility for the deaf and hearing-impaired communities.

### *A. Dataset Preparation*

Collecting a sign language image dataset for ASL and ISL involves several steps. Firstly, a large number of images are collected that feature sign language gestures from both ASL and ISL. These images can be collected from various sources, such as online image repositories or by taking photographs of sign language users. Table I shows the details of the dataset.

Subsequently, a crucial step involves the annotation of all images with corresponding labels that precisely identify the specific sign language gesture being executed. The resulting dataset is then divided into distinct subsets for training, validation, and testing purposes, as detailed in Table II. The training is performed by using train set along with fine tuning with validation set from epoch to epoch. Ultimately, the model's performance is evaluated using the testing set to determine its final effectiveness

TABLE I Collected Sign Language Dataset

| Dataset | Number of Images for Alphabets | Number of Images for Numbers | Images in Each class | Total Images |
|---|---|---|---|---|
| American Sign Language Dataset [8] | 1820 | 700 | 70 | 2520 |

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

| | | | | |
|---|---|---|---|---|
| Indian Sign Language Dataset [9] | 31200 | 12000 | 1200 | 43200 |

TABLE II Dataset Preparation for Experimentation

| **Dataset** | **Selected Images in Each class** | **Total Images (X)** | **Testing Set (20 % of X)** | **Train Set ($X_1$ = 80% of X)** | |
|---|---|---|---|---|---|
| | | | | **Training Set (80% of $X_1$)** | **Validation Set (20% of $X_1$)** |
| American Sign Language Dataset | 70 | 2520 | 504 | 1612 | 404 |
| Indian Sign Language Dataset | 120 | 4320 | 864 | 2766 | 690 |

### *B. CNN Model Development*

Attention Layer:

The attention layer is composed of parallel combination of Maxpooling and average pooling layer [10]. The proposed attention net model consist of Convolution layered architecture similar to VGG16 pattern. The pattern consist of two convolution and one max pooling layer. The concatenation layer is added which concatenates the features from the attention layer and convolution layer output. The proposed Attention based CNN is shown in Figure 2. The fusion approach is shown in block diagram in Figure 3.

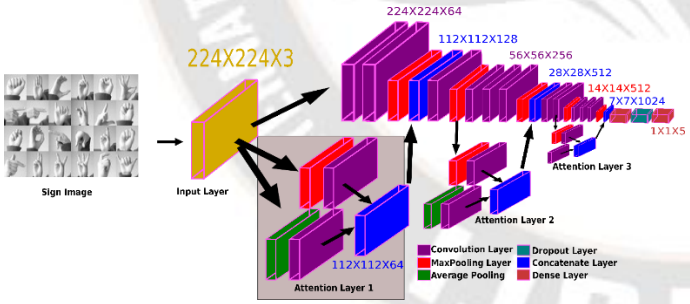



Figure 2: Proposed Attention Based Model

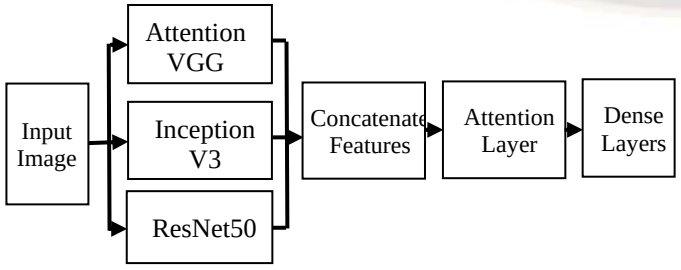



Figure 3: Fusion of Models with Transfer Learning

Residual Block:

Residual blocks constitute a foundational element in the architecture of ResNets [11]. Initially introduced by He et al. [11], these blocks have garnered substantial attention and are now integral components in various state-of-the-art models. The difference in input and output layers is significantly captured with these blocks. The skip connection in this blocks this role of extracting relevant features. The schematic representation of a residual block is illustrated in Figure 4. The input is passed through sequential processing steps of convolutional layer, batch normalization and a Rectified Linear Unit (ReLU) activation. Additionally, convolutional layer and batch normalization are applied, which is then clubbed with output of skipped connection. The concatenated output is then subjected to another ReLU activation function before being presented as the output of the residual block. This way the network prioritizes optimizing the residual function over the initial mapping from input to output, effectively addressing the challenge of vanishing gradients across multiple layers.

The skip connection can be represented mathematically as follows:

$$y = activation\left(input + Conv2D(3\times 3)\left(activation\left(Conv2D(3\times 3)(input)\right)\right)\right) \quad ...(1)$$

Where the residual function combination is learned by the block over the input to get the output y.

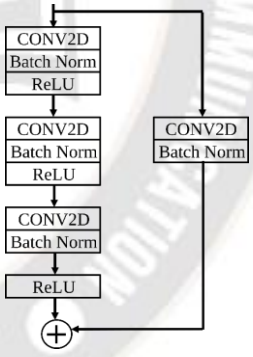



Figure 4: Residual Block

Bottleneck Block

The bottleneck block is a variant of the conventional residual block commonly utilized image recognition purposes in CNNs. The primary objective of incorporating the bottleneck block in a CNN is to decrease the parameter count while still allowing the network to acquire complex representations of the input data. With kernel configurations of 1x1, 3x3 and then again 1x1 of 3 convolutional layers constitute the bottleneck block. This combination reduces the dimensionality of the feature vector. The architectural representation of a bottleneck block can be observed in Figure 5.

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

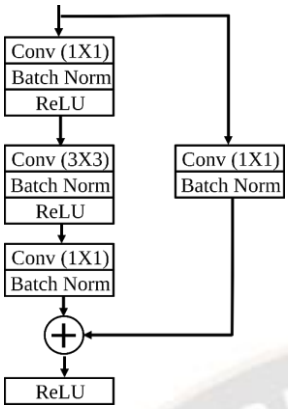



Figure 5: Bottleneck Block

Within this particular architecture, the initial 1x1 convolution operation serves to decrease the quantity of input channels by a factor of 4, while the subsequent 1x1 convolution operation increases the number of output channels to align with the original count. This reduction and subsequent augmentation of channel numbers contribute to the reduction in computational complexity of the subsequent 3x3 convolutional layer. Simultaneously, it ensures that the network retains the capacity to acquire intricate and complex features during the learning process. Similar to the standard residual block, the bottleneck block also incorporates a skip connection, which operates in the same manner. This skip connection enables the network to learn residual functions and propagate information through shortcut connections. By doing so, the network can effectively leverage the benefits of residual learning and enhance the flow of information within the architecture.

The mathematical equation for a bottleneck block can be written as follows:

$$y = Concat(Conv2d(1 \times 1)(input), Conv2D(3 \times 3)(input)) \quad ...(2)$$

Here, y represents the output learned over input with bottleneck block. The residual function F encompasses the 1x1 and 3x3 convolutions, along with the corresponding activation functions. The output y is obtained by summing the residual function and the input, followed by an activation function. The bottleneck block serves as a valuable tool for reducing the computational complexity of deep learning models while enabling them to learn intricate representations of input data. By employing 1x1 convolutions to reduce and increase channel numbers, and a 3x3 convolution for feature extraction, the bottleneck block facilitates efficient and effective deep learning architectures. In 2014, Google introduced InceptionNet (GoogLeNet), a renowned CNN architecture featuring innovative inception modules. These modules enable diverse feature learning at various scales with low parameters. InceptionNet tackles challenges of deeper networks by combining convolutional layers, pooling, and parallel paths. Excelling in image classification and object detection, its influence led to Google's 2015 Inception v3, an improved version enhancing accuracy and efficiency over the original InceptionNet.

The main improvements in Inception V3 compared to the original InceptionNet are:

- Factorization of convolutions: Inception V3 employs factorized convolutions, splitting a standard convolution into two operations: 1xN followed by Nx1. This reduces parameters and enhances computational efficiency in the network.
- Reduction in spatial resolution: Inception V3 employs pooling and convolutional layers to decrease feature map spatial resolution, lowering computational costs and enhancing model accuracy.
- Use of batch normalization: Inception V3 uses batch normalization, which helps to improve the convergence rate and reduce the impact of covariance of the network.
- Improved regularization: Inception V3 uses various regularization techniques, including dropout and L2 regularization, which reduces overfitting of the model.
- Reduction in number of parameters: Inception V3 minimizes parameters by using techniques like convolutions factorization and reducing filter count in certain convolutional layers.

Combining the features extracted by ResNet50 [11] and Inception V3 [12] through concatenation and passing them through an attention layer that consists of parallel combinations of convolution, max pooling, and average pooling layers could potentially improve the performance of the model, particularly in tasks where multiple sources of information need to be integrated to make a decision. The concatenation of the features extracted by ResNet50 and Inception V3 would increase the richness of the feature representation, as each network is known to extract different types of features. The attention layer would then allow the model to selectively focus on the most informative features, by learning a set of attention weights that are applied to the features.

The concurrent integration of convolution, max pooling, and average pooling layers within the attention layer enables the model to grasp diverse facets of features. This includes aspects like spatial distribution, maximum and average values, as well as frequency content. By combining these different aspects, the attention layer could help the model identify the most relevant features for the task at hand.

Let $x_{resnet}$ and $x_{inception}$ be the feature maps extracted by ResNet50 and Inception V3, respectively. The concatenated feature map can be represented as:

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

$$x_{concat} = Concatenate([x_{resenet}, x_{inception}])$$

*...( 3)*

where Concatenate is the concatenation layer in Keras or other deep learning frameworks.

The attention weights w can be learned using a dense layer with softmax activation, applied to a feature map y:

$$w = Dense(units = 2, activation = 'softmax')(y)$$

*...( 4)*

where units=2 represents the two possible attention weights (one for ResNet50 and one for Inception V3), and softmax ensures that the weights sum up to one.

The attention layer can be defined as:

$$y = Concatenate\left(\begin{bmatrix} Conv2D(filters = 16, kernel_{size} = (3,3), padding = 'same', activation = 'relu')(x_{concat}), \\ MaxPooling2D(pool_{size} = (2,2))(x_{concat}), \\ AveragePooling2D(pool_{size} = (2,2))(x_{concat}) \end{bmatrix}\right)$$

*...( 5)*

Where Conv2D, MaxPooling2D, and AveragePooling2D represent the convolution, max pooling, and average pooling layers, respectively. The multiply layer applies the learned attention weights to the feature maps, and the Add layer combines the attention-weighted features with the original features.

Spatial Attention in CNN Model

Spatial attention mechanisms in CNNs enable the model to selectively focus on certain regions of the input image or feature map based on their relevance to the task at hand. One way to implement spatial attention is to learn a set of spatial attention weights. The element wise multiplication is performed while applying these weights to the feature map. Here's a more detailed explanation of how spatial attention can be implemented in CNNs using mathematical equations:

For input feature map x (H × W × C), spatial attention weights a (H × W × 1) are learned via a CNN. Using element-wise multiplication, the attention-weighted feature map y is obtained:

$$y = a * x$$

*...( 6)*

Where * represents element-wise multiplication. The attention-weighted feature map y can then be passed through additional convolutional layers and pooling layers to extract higher-level relevant features. To ensure that the attention weights sum up to one, we can normalize them using the softmax function:

$$a' = softmax(a)$$

*...( 7)*

Where a' is the normalized attention weights. Finally, original feature map x is applied with normalized attention weights to get the attention-weighted feature map y. Thus,

$$y = a' * x$$

*...( 8)*

Where * represents element-wise multiplication. The attention weights can be learned using a small CNN, and they are typically normalized using the softmax function to ensure that they sum up to one. The resulting attention-weighted feature map can then be passed through additional layers to extract higher-level relevant features.

Combination for Attention Layer

The combination of spatial and channel attention mechanisms in CNNs involves learning both spatial and channel attention weights, and applying them to the feature map using element-wise multiplication. Here's a more detailed explanation of how this approach can be implemented in CNNs using mathematical equations:

Input feature map x (H × W × C). Spatial attention weights as (H × W × 1) and channel attention weights ac (C × 1) are learned:

$$y = (a_s * a_c) * x$$

*...( 9)*

Where * represents element-wise multiplication. The attention-weighted feature map, denoted as y, can be subsequently subjected to further processing through additional convolutional layers and pooling layers. These subsequent layers aim to extract higher-level features that hold greater relevance to the specific task being performed. To ensure that the attention weights sum up to one, we can normalize the spatial attention weights using the softmax function, and normalize the channel attention weights using the sigmoid function:

$$a'_s = softmax(a_s)$$
$$a'_c = sigmoid(a_c)$$

*...( 10)*

Where $a'_s$ is the normalized spatial attention weights, and $a'_c$ is the normalized channel attention weights. Finally, the attention-weighted feature map y can be obtained by applying the normalized attention weights to the original feature map x:

$$y = (a'_s * a'_c) * x$$

*...( 11)*

Where * represents element-wise multiplication. The combination of spatial and channel attention in CNNs involves learning both spatial and channel attention weights using small convolutional neural networks. The attention weights are applied to the input feature map using element-wise multiplication, and the resulting attention-weighted feature map is passed through additional layers to extract higher-level

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

features that are more relevant to the task at hand. The attention weights are typically normalized using the softmax function for spatial attention, and the sigmoid function for channel attention, to ensure that they sum up to one and are positive.

## III. Results and Discussion

If separate analyses were conducted on ASL and ISL datasets by training at different sessions, it suggests that the datasets were not combined, and the models were trained on each dataset separately. Training on separate sessions means that the model was trained on different occasions or time periods, which could have led to variations in the model's performance. However, training on different sessions can also be beneficial as it can help reduce overfitting to specific instances or batches of data, leading to more robust models. It is important to note that ASL and ISL are distinct sign languages with their own unique features, and training separate models for each language is appropriate for achieving high accuracy in recognizing signs in each language.

### *A. Performance Parameters*

The performance of Model is evaluated using Accuracy, Sensitivity, Specificity and F1 Score parameters. The formulae for these parameters are shown in Table III.

TABLE III Performance Parameters

| Parameter | Formula |
|---|---|
| Accuracy | TP+TN/(TP+TN+FP+FN) |
| Specificity | TN/(TN+FP) |
| Sensitivity(Recall) | TP/(TP+FN) |
| Precision | TP/(TP+FP) |
| F1 Score | 2*(Recall*Precision)/(Recall+Precision) |

In sign language recognition, TP, TN, FP, and FN are terms used to evaluate the accuracy of a machine learning model's predictions.

- In TP (True Positive) cases, a sign language gesture is correctly identified with respect to ground truth.
- In TN (True Negative) cases, a gesture is recognized that does not belong to a particular category or class.
- In FP (False Positive) cases, a gesture is recognized as it belonging to a particular category or class, when in fact it does not.
- In FN (False Negative) cases, a gesture is recognized as not belonging to a particular category or class, when in fact it does.

In general, a good sign language recognition model should have a high TP rate and a low FP rate, indicating that it correctly identifies signs belonging to the desired category, while avoiding false alarms. The ultimate goal is to maximize both TP and TN rates while minimizing FP and FN rates.

### *B. 10-Fold Analysis*

A commonly employed technique for cross-validation in machine learning is the 10-fold analysis [13]. It involves partitioning the dataset into 10 equally-sized subsets, training the model on nine of these subsets, and utilizing the remaining subset for validation. This process is repeated 10 times, ensuring that each subset serves as the validation set once. By adopting this approach, a more reliable estimation of the model's ability to generalize to new and unseen data can be obtained, in contrast to training and validating on the same dataset. Applying the 10-fold analysis to both ASL and ISL datasets can enhance the model's accuracy, generalization, and provide valuable insights into the distinctive characteristics of each language. Figure 6 and Figure 7 depict the accuracy results obtained from the 10-fold analysis for the ASL and ISL datasets, respectively.

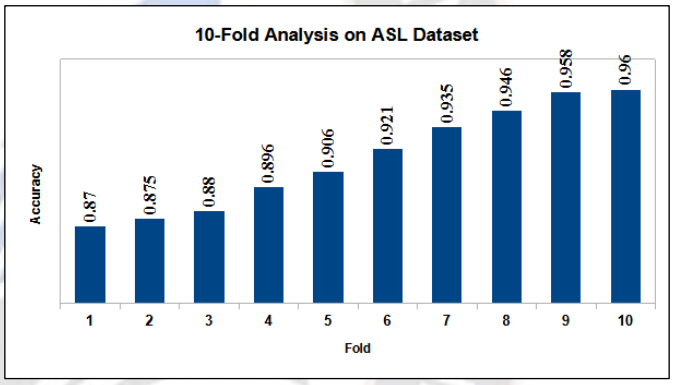



Figure 6: 10-Fold Analysis on ASL Dataset

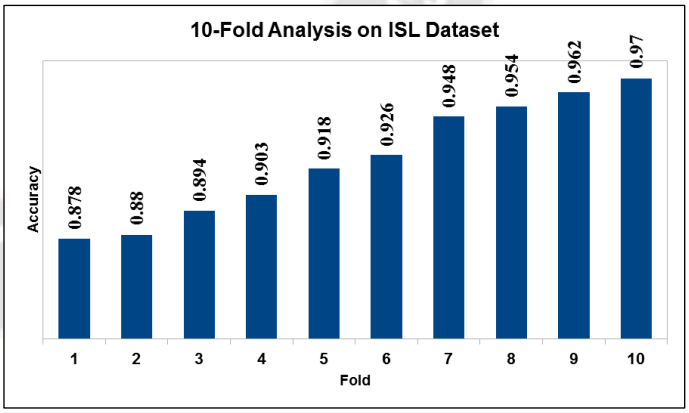



Figure 7: 10-Fold Analysis on ISL Dataset

### *C. Performance of Model on ASL and ISL dataset*

Figure 8 and Figure 9 show the performance graphs of model with different combinations. The performance of models AttentionNet (AN), ResNet50 (RN) and Inception (V3) are compared in different combinations with proposed hybrid

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:*** *18 August 2023* ***Revised:*** *21 September 2023* ***Accepted:*** *16 October 2023*

model (AN+RN+IN). The combination of three models show highest performance with accuracy of 0.96 on ASL and 0.97 on ISL datasets respectively.

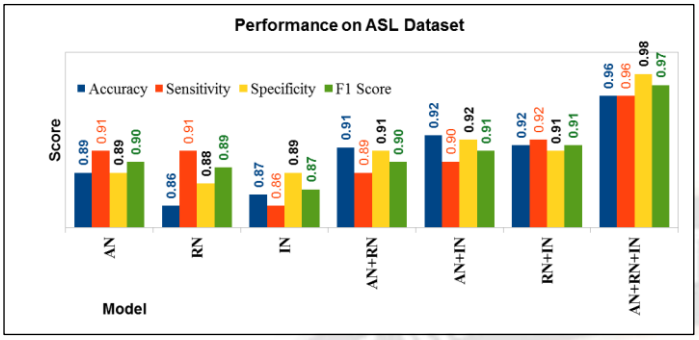



Figure 8: Performance Evaluation on ASL Dataset

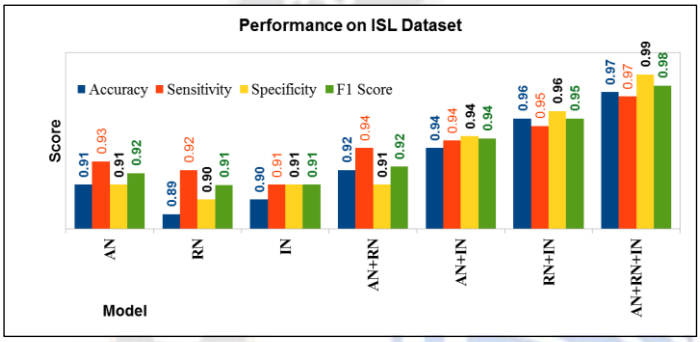



Figure 9: Performance Evaluation on ISL Dataset

### *D. Comparative with State-of-the-art Models*

VGG16 [14], ResNet101 [11], DenseNet201 [15], and MobileNet-V2 [16] are all highly advanced deep learning models that have demonstrated exceptional performance across diverse computer vision tasks. VGG16, developed by the Visual Geometry Group (VGG) at Oxford University, is a convolutional neural network (CNN) architecture. It comprises 16 layers and is renowned for its simplicity and ease of implementation. VGG16 has delivered remarkable results in image classification tasks, serving as a baseline model in numerous research studies. ResNet101, introduced by Microsoft Research in 2015, represents a deep residual network. With 101 layers, it addresses the challenge of vanishing gradients encountered in exceedingly deep neural networks. ResNet101 has achieved better performance in computer vision based classification tasks. DenseNet201, developed by researchers at Facebook AI Research in 2017, is a densely connected convolutional neural network. With 201 layers, it stands out for its efficiency and accuracy. DenseNet201 has delivered exceptional performance in multiple computer vision tasks, including image classification, object detection, and semantic segmentation. MobileNet-V2, unveiled by Google in 2018, is a lightweight convolutional neural network. Designed to be highly efficient and fast, it particularly suits mobile and embedded devices. MobileNet-V2 has attained state-of-the-art performance in diverse computer vision tasks while demanding significantly fewer computational and memory resources compared to larger models.

The selection of these models holds significance as they represent various architectures and design choices that have proven effective across different computer vision tasks. By comparing the performance of these models on a specific application task, valuable insights can be gained regarding the strengths and weaknesses of different deep learning architectures. This information can further inform the development of new models and advancements in the field. Figure 10 and Figure 11 show the comparative analysis of proposed method with other state-of-the art models.

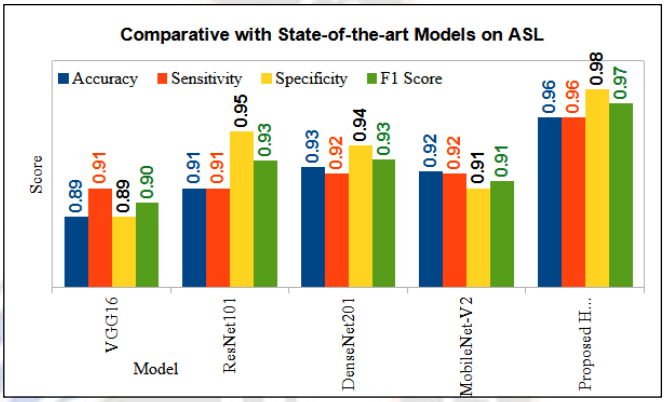



Figure 10: Comparative with State-of-the-art Models on ASL Dataset

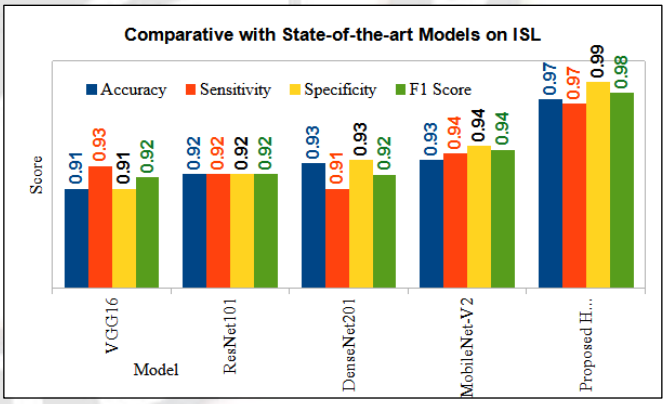



Figure 11: Comparative with State-of-the-art Models on ISL Dataset

### *E. Discussion*

In a study by Prasath et al. [17], a combined encoder with a multi-layered CNN model was proposed for Sign Language Recognition (SLR). They utilized the ROBITA ISL database as input and implemented essential pre-processing steps to ensure accurate analysis. The encoder, known as Multi-Layer CNN (ML-CNN), was designed for better performance of the SLR system. By analyzing both linear and non-linear features, the encoder aimed to enhance recognition quality. The ML-CNN

*International Journal on Recent and Innovation Trends in Computing and Communication*
*ISSN: 2321-8169 Volume: 11 Issue: 10*
***Article Received:** 18 August 2023 **Revised:** 21 September 2023 **Accepted:** 16 October 2023*

model with the encoder achieved a prediction accuracy of 87.5% with better performance over BLSTM and HMM models. The performance is boosted by 1% and 3.5% respectively. In another work by Enireddy et al. [18], proposed a CNN model for recognizing baby sign language. They utilized an infant sign language dataset and conducted several pre-processing steps, including frame extraction from videos and data augmentation techniques. The Enhanced CNN (ECNN) was employed to extract features from the frames, and a novel optimization algorithm called Life Choice Based Optimizer (LCBO) was used for selecting optimal characteristics. Finally, the Deep LSTM scheme was employed for classification. The ECNN-DLSTM approach achieved an impressive accuracy of 99% and a kappa value of 96%, surpassing various deep and machine learning approaches. Guo et al. [19] shown effective strategy of spatiotemporal features modelling with a global-local representation (GLR) module. The temporal dimensions with respect to height and width of the feature map are extracted using GLR using local and global shifts. Local path based detailed features are extracted in local shifts based processing approach. A holistic representation of features is seen in global shifts. The information with different dimensions is collected using a cross-scale aggregation module. The performance of the model is evaluated on three benchmarks WLASL, INCLUDE, and LSA64. Table IV in the research papers presents a comparative analysis of the existing methods along with the proposed method, showcasing the performance differences between the approaches.

TABLE IV Comparative Performance Analysis of Sign Language Recognition Methods

| Method | Dataset | Classes | Performance |
|---|---|---|---|
| Custom CNN [17] | ROBITA ISL | 35 | Accuracy 87.5% |
| ECNN [18] | Baby Sign Language | 15 | Accuracy 99% |
| global–local representation (GLR) module [19] | WLASL, INCLUDE and LSA64, | 35 | Accuracy 98% |
| The multi-view spatiotemporal embedding network (MSTEN), the continuous sign language recognition network (CSLRN), and the sign language translation network (SLTN) are three distinct networks used in the field of sign language processing [20] | SLR-100, RWTH, and CSL-daily | 35 | Accuracy 98% |
| Proposed | ASL | 35 | Accuracy 97% |
| | ISL | 35 | Accuracy 98% |

## IV. CONCLUSION

The sign language recognition model, meticulously crafted for optimal performance, has demonstrated extraordinary accuracy, marking a significant breakthrough in communication for the hearing-impaired community. With a stellar accuracy rate of 0.96 for American Sign Language (ASL) and 0.97 for Indian Sign Language (ISL), coupled with impressive specificity, sensitivity, and F1 scores, the model excels in deciphering sign languages accurately.

This success can be attributed to the model's innovative hybrid approach, seamlessly integrating hand-crafted techniques and deep learning methodologies. The hand-crafted aspect employs predefined features and rules to interpret sign language, while the deep learning component harnesses neural networks to learn from data and make precise predictions. The synergy between these approaches contributes to the model's exceptional accuracy and resilience against variations and deformations in signs. Notably, the hybrid model's accuracy outshines previous models, which typically achieved rates around 80-85%. This leap in accuracy is pivotal, rendering the technology more practical and accessible for the hearing-impaired community. The model's reliability makes it viable for real-world applications, particularly in educational and medical settings, facilitating improved communication between hearing-impaired individuals and others. In essence, the hybrid sign language recognition model emerges as a potent force in advancing communication between individuals with hearing impairments and the broader community, underscoring its transformative potential in the realm of sign language recognition.

**Competing Interests:**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Author's contribution statement**

The paper background work, conceptualization, methodology, Dataset collection, implementation, result analysis and comparison, preparing and editing draft, visualization, review of work and project administration, have been done by first author under the supervision of second author.

**Ethical and informed consent for data used**

Written informed consent was obtained from the sources of datasets.

**Data Availability and access:**

American Sign Language dataset is available at: https://www.kaggle.com/datasets/prathumarikeri/american-sign-language-09az

Indian Sign Language Dataset is available at: https://www.kaggle.com/datasets/prathumarikeri/indian-sign-language-isl

REFERENCES

[1] H. Cooper, B. Holt, and R. Bowden, "Sign Language Recognition," in Visual Analysis of Humans, Springer, London, 2011, pp. 539–562.

[2] M. Marschark, E. Machmer, L. J. Spencer, G. Borgna, A. Durkin, and C. Convertino, "Language and Psychosocial Functioning among Deaf Learners with and without Cochlear Implants," J. Deaf Stud. Deaf Educ., vol. 23, no. 1, p. 28, Jan. 2018, doi: 10.1093/DEAFED/ENX035.

[3] M. M. Balaha *et al.*, "A vision-based deep learning approach for independent-users Arabic sign language interpretation," *Multimed. Tools Appl.*, vol. 82, no. 5, pp. 6807–6826, Feb. 2023, doi: 10.1007/S11042-022-13423-9/TABLES/6.

[4] I. Papastratis, C. Chatzikonstantinou, D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Artificial Intelligence Technologies for Sign Language," *Sensors (Basel).*, vol. 21, no. 17, Sep. 2021, doi: 10.3390/S21175843.

[5] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, "Advances in machine translation for sign language: approaches, limitations, and challenges," *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14357–14399, Nov. 2021, doi: 10.1007/S00521-021-06079-3/METRICS.

[6] "What Is American Sign Language (ASL)? | NIDCD." https://www.nidcd.nih.gov/health/american-sign-language (accessed May 10, 2023).

[7] "ISL (Indian Sign Language Portal) || FDMSE || RKMVERI || COIMBATORE CAMPUS – Dictionary on Indian Sign Language (ISL) signs by FDMSE, Coimbatore." https://indiansignlanguage.org/ (accessed May 10, 2023).

[8] "American Sign Language Dataset | Kaggle." https://www.kaggle.com/datasets/ayuraj/asl-dataset (accessed May 10, 2023).

[9] "Indian Sign Language Dataset | Kaggle." https://www.kaggle.com/datasets/vaishnaviasonawane/indian-sign-language-dataset (accessed May 10, 2023).

[10] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11211 LNCS, pp. 3–19, 2018, doi: 10.1007/978-3-030-01234-2_1/TABLES/8.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, Dec. 2015, doi: 10.1109/CVPR.2016.90.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, Dec. 2015, doi: 10.1109/CVPR.2016.308.

[13] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," *Encycl. Database Syst.*, pp. 532–538, 2009, doi: 10.1007/978-0-387-39940-9_565.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, Sep. 2014, Accessed: May 10, 2023. [Online]. Available: https://arxiv.org/abs/1409.1556v6.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, Aug. 2016, doi: 10.1109/CVPR.2017.243.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, Jan. 2018, doi: 10.1109/CVPR.2018.00474.

[17] G. Arun Prasath and K. Annapurani, "Prediction of sign language recognition based on multi layered CNN," *Multimed. Tools Appl.*, pp. 1–21, Mar. 2023, doi: 10.1007/S11042-023-14548-1/METRICS.

[18] V. Enireddy, J. Anitha, N. Mahendra, and G. Kishore, "An optimized automated recognition of infant sign language using enhanced convolution neural network and deep LSTM," *Multimed. Tools Appl.*, pp. 1–23, Feb. 2023, doi: 10.1007/S11042-023-14428-8/METRICS.

[19] Z. Guo, Y. Hou, and W. Li, "Sign language recognition via dimensional global–local shift and cross-scale aggregation," *Neural Comput. Appl.*, pp. 1–13, Mar. 2023, doi: 10.1007/S00521-023-08380-9/METRICS.

[20] R. Li and L. Meng, "Sign language recognition and translation network based on multi-view data," *Appl. Intell.*, vol. 52, no. 13, pp. 14624–14638, Jul. 2022, doi: 10.1007/S10489-022-03407-5/METRICS.

)