# Static and Dynamic Isolated Indian and Russian Sign Language Recognition with Spatial and Temporal Feature Detection Using Hybrid Neural Network



E. RAJALAKSHMI and R. ELAKKIYA, School of Computing, SASTRA Deemed University, Thanjavur, India
ALEXEY L. PRIKHODKO, M. G. GRIF, and MAXIM A. BAKAEV, Novosibirsk State Technical University, Russian Federation
JATINDERKUMAR R. SAINI, Symbiosis Institute of Computer Studies and Research, Symbiosis International (Deemed University), Pune, India
KETAN KOTECHA, Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, India
V. SUBRAMANIYASWAMY, School of Computing, SASTRA Deemed University, Thanjavur, India

The Sign Language Recognition system intends to recognize the Sign language used by the hearing and vocally impaired populace. The interpretation of isolated sign language from static and dynamic gestures is a difficult study field in machine vision. Managing quick hand movement, facial expression, illumination variations, signer variation, and background complexity are amongst the most serious challenges in this arena. While deep learning-based models have been used to accomplish the entirety of the field's state-of-the-art outcomes, the previous issues have not been fully addressed. To overcome these issues, we propose a Hybrid Neural Network Architecture for the recognition of Isolated Indian and Russian Sign Language. In the case of static gesture recognition, the proposed framework deals with the 3D Convolution Net with an atrous convolution mechanism for spatial feature extraction. For dynamic gesture recognition, the proposed framework is an integration of semantic spatial multi-cue feature detection, extraction, and Temporal-Sequential feature extraction. The semantic spatial multi-cue feature detection and extraction module help in the generation of feature maps for Full-frame, pose, face, and hand. For face and hand detection, GradCam and Camshift algorithm have been used. The temporal and sequential module consists of a modified auto-encoder with a GELU activation function for abstract high-level feature extraction and a hybrid attention layer. The hybrid

This work was supported by the Department of Science & Technology (DST), India for the financial support through the Indo-Russian Joint Project (INT/RUS/RFBR/393). We also acknowledge SASTRA Deemed University, Thanjavur, India, for extending infrastructural support to carry out this research work.
Authors' addresses: E. Rajalakshmi and R. Elakkiya (corresponding author), School of Computing, SASTRA Deemed University, Thanjavur 613401, India; emails: rajalakshmi32210@gmail.com, elakkiyaceg@gmail.com; A. L. Prikhodko, M. G. Grif, and M. A. Bakaev, Novosibirsk State Technical University, 20, K. Marx Prospekt, Novosibirsk, 630073, Russian Federation; emails: alexeyayay@yandex.ru, grif@corp.nstu, bakaev@corp.nstu.ru; J. R. Saini, Symbiosis Institute of Computer Studies and Research, Symbiosis International (Deemed University), Pune, India; email: saini_expert@yahoo.com; K. Kotecha, Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, India; email: drketankotecha@gmail.com; V. Subramaniyaswamy, School of Computing, SASTRA Deemed University, Thanjavur, India; email: subramaniyaswamy@gmail.com.

2375-4699/2022/11-ART26 $15.00
https://doi.org/10.1145/3530989

attention layer is an integration of segmentation and spatial attention mechanism. The proposed work also involves creating a novel multi-signer, single, and double-handed Isolated Sign representation dataset for Indian and Russian Sign Language. The experimentation was done on the novel dataset created. The accuracy obtained for Static Isolated Sign Recognition was 99.76%, and the accuracy obtained for Dynamic Isolated Sign Recognition was 99.85%. We have also compared the performance of our proposed work with other baseline models with benchmark datasets, and our proposed work proved to have better performance in terms of Accuracy metrics.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; **Computer vision**; **Computer vision tasks**; **Activity recognition and understanding**;

Additional Key Words and Phrases: Isolated sign language recognition, convolutional neural network, bidirectional LSTM, Indian sign language, gesture recognition

**ACM Reference format:**
E. Rajalakshmi, R. Elakkiya, Alexey L. Prikhodko, M. G. Grif, Maxim A. Bakaev, Jatinderkumar R. Saini, Ketan Kotecha, and V. Subramaniyaswamy. 2022. Static and Dynamic Isolated Indian and Russian Sign Language Recognition with Spatial and Temporal Feature Detection Using Hybrid Neural Network. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 22, 1, Article 26 (November 2022), 23 pages.
https://doi.org/10.1145/3530989

## 1 INTRODUCTION

Deaf and mute populations use **Sign Language (SL)** as a primary means of interaction. According to the WHO, 6.1% of the populace are deaf or have a speech handicap. Based on the **National Association of the Deaf (NAD)** reports, approximately 18 million Indians have impaired hearing. The sign language used in different parts of the world is different. There are more than 120 different SLs. Arabic [1], American, Bhutanese [2], German, Chinese [3], Russian, Turkish [4], and Indian sign languages [5, 61] are just a few of the well-known sign languages. Recent investigations on SL comprehension [6, 7] have garnered considerable attention. Some studies included recognition with iterative training [8], while some involved temporal segmentation [9]. Many other pieces of research were carried out on SL using a combination of CNN, LSTM, and HMM [10]. Some existing systems have used Grassman Covariance matrix [11], or even iterative alignment net, [12] transformers [13, 60], activity recognition techniques [58, 59], attention mechanisms, and so on. Though the existing Sign Language Recognition frameworks cover some of the challenges, they still do not completely address some major challenges such as integration manual and non-manual feature extraction with multi-signer datasets and unavailability of large open-source SL datasets with a large vocabulary, handling variation in signer complexion, illumination condition, and so on. Moreover, most frameworks leverage physical sensors, depth sensor cameras, and so on, to record the sign dataset, making it quite expensive and uncomfortable for signers to perform the sign gestures. We have implemented a vision-based Sign Language Recognition System without any wearable or physical sensors to overcome such constraints, making the framework affordable and signer-friendly.

This article will focus on the **Isolated Sign Language Recognition System (ISLR)**. The main objective of the ISLR is to recognize the static and isolated sign representation of words and alphanumeric characters and hence classify them. Sign language involves manual elements from the hands and non-manual elements from the face and upper-body posture to efficiently and properly represent the desired notion [14]. Both hands' shape, position, orientation, and movement are considered manual aspects, but the eye glance, mouth shape, facial expression, and body poise are non-manual elements. Without any effort, the human visual perception empowers us to process

and understand this instantaneous yet complicated information. However, a neural network needs to strive to detect the tacit collaboration of several visual features completely without specialized expertise. Most **Sign Language Recognition (SLR)** systems use external sensors or gloves, which is quite uncomfortable as it restricts movement. Hence the more natural and affordable SLR system for the common person would be the Video-based SLR systems. When it comes to Video-based SLR systems, there are many challenges faced recognizing Sign Language. The signer may have varying skin tones, and the signer may have complex backgrounds varying illumination conditions. While expressing the sign gestures, some signers may be more expressive, and some may not. Also, the standard spoken language has a large vocabulary, and considering the day-to-day general conversational words for sign language is the biggest challenge. The lack of a publicly available dataset for such a large vocabulary makes it quite difficult to build a full-fledged SLR for the hearing impaired community. Most of the studies have not considered these natural conditions while framing the SLR systems as the recognition task becomes more and more complex while we start including all the features.

To overcome these above-stated issues, we have proposed a Hybrid Neural Network framework for recognizing Isolated Indian and Russian Sign Language Words and Alphabets. In this proposed work, we have created our isolated sign datasets. This novel dataset is partially available in a public repository for further research. The created dataset includes both Static and Dynamic gestures and Single-handed and Double-handed gestures. The dataset consists of sign gesture images and videos captured with the help of multiple signers having varying skin tones, complex backgrounds, and varying lighting conditions. The dynamic sign representations were extracted to frames. The Dynamic frames and the static images were then processed and passed through the proposed **Hybrid Neural Network framework for Isolated Sign Language Recognition (HNN-ISLR)**. The proposed HNN-SLR combines Semantic Spatial and Temporal feature extraction using an integration of Convolutional Neural Net and Bidirectional LSTM. The sign gestures were first inputted to the semantic spatial feature extraction module, wherein the manual and non-manual features were extracted from the mainframe image using GradCam and Camshift. The skeletal pose was also extracted from the frame. The semantic features were extracted from the Full-frame, pose, face, and hand in the semantic spatial feature extraction phase. For extracting the temporal features, Modified Auto-encoder with GELU [15] activation function was leveraged to extract more high-level abstract features. A Hybrid Attention layer was developed incorporating the segmentation and spatial attention mechanism to distinguish between the genuine and non-useful sign gestures. For the recognition and classification of the Sign language gestures, a BiLSTMNet was constructed. At last, a fully connected layer with the softmax function was used to formulate classification probabilities. The proposed work has the following objectives:

(1) Creation of a novel dataset for Indian and Russian ISLR.
(2) Implementation of spatial semantic feature extraction using 3D Conv Net using Atrous Convolution for Static Sign Representation.
(3) Implementation of spatial multi-semantic feature extraction framework for extracting semantic manual and non-manual features from the sign gestures, thereby creating feature maps for Full-frame, Hand, Face, and Pose for Dynamic Sign Representations.
(4) The Temporal and Sequential feature extraction feature is implemented to extract more abstract features using modified auto-encoders and provide weightage to genuine and non-useful gestures using a hybrid attention layer.
(5) Construction of BiLSTMNet and fully connected layer and softmax function for final recognition and classification of Isolated Sign gestures. Performance evaluation of proposed framework with own dataset and comparison with other benchmark datasets.

This article is further organized into several sections. Section 2 discusses the literature survey and the related works. The Proposed Method is discussed in Section 3. Section 4 deals with the results and discussion, and Section 5 discuss the conclusion and future works.

## 2 LITERATURE SURVEY

There have been many studies going on in Sign Language Recognition. Different models for determining the composition of signs in terms of subunits similar to a phoneme in a verbal language have already been reported in various studies [16, 17]. Subunits of a sign language can be characterized using the Stokoe model based on the shape, orientation, and mobility of the hand [18]. The fingers' configuration and the palm's alignment explain the handshape, while the placement of the hand about the body is determined. The Movement-Hold model [19] has been considered to consist of sequential organization of movement, and static posture is continuous signing. The models mentioned above are satisfactory for one-handed signs. However, in two-handed signs, the relationship between the movement and postures of the two hands has been simulated as well [20]. Sign Language Recognition can be categorized into sensor-based (multi-modality) and vision-based (multi-semantic). Physical sensors [21, 22] were used in the initial works on multi-modality to capture three-dimensional space knowledge, including infrared maps and depth [23, 24]. A few works have investigated the multi-modality fusion of optical flow and RGB that yields state-of-the-art performances on the PHOENIX-2014 database. There are several challenges in developing an SLR. Even if the same person makes the identical sign twice, slight hand velocity and position differences might occur.

There are also several challenging difficulties, such as hand tracking or hand segmentation from environment and context, differences in illumination, occlusion, and position [25]. The majority of isolated sign recognition investigation has eased the difficulty of segmentation and tracking by using equipment on the hands, such as colour gloves or markers, to detect position features directly. Another framework was proposed, with Sign tutor stages for hand and face detection, analysis, and classification [26]. This framework used coloured gloves for easy hand segmentation and detection. Another proposed method employed colour-coded gloves to make human hand tracking easier. Algorithms retrieve major characteristics from videos using a multi-colour detection method and a Hidden Markov Model for classification. A multi-function extraction was proposed that uses the hand as the only recognized object [27]. These features include the colour histogram, Hue moments, Gabor wavelet, Fourier descriptor, SIFT features, and classification support vector machine. With a 90% accuracy, another author proposed an approach for recognizing static, isolated signs using a directional histogram and classification using Euclidean Distance and K-nearest neighbour [28]. An SLR was proposed with an Optimized Neural Network for efficient recognition [29]. The open pose was used for another model to extract the posture cue of the Sign gesture [30]. Another SVM and skin segment-based model were proposed in Reference [31].

A novel approach was proposed for SLR that considered the complex background scenario with dynamic sign gesture [32]. In one of the proposed works, a dynamic sign representation recognition system was also introduced for Indian Sign Language [33]. A grid-based method was introduced using Indian Sign Language [34]. Another Indian Sign Language Recognition system was developed using Bayesian KNN-based sign recognition to remove gesture redundancy [35]. Some researchers also considered the differentiation of manual and non-manual features for SL recognition [36]. A pose-based word-level SLR was also developed [37]. A Sign language translation framework was developed for pre-trained and end-to-end settings using Neural Machine Translation [38]. In another study, the author had proposed a temporal accumulative feature extraction for developing ISLR.

Hand energy and modelling-based ISLR were developed using Convolution Neural Network to extract the manual features of the sign gestures [39]. A novel signer-independent ISLR system was introduced, DeepArSLR, to address the multi-signer issues using Deep Learning for Arabic SL [40]. A word-level SLR wherein the proposed model learned the domain invariant descriptors [41]. Another model incorporated the Deep Cascade model for Spatio-temporal video-based ISLR [42]. A multi-modal SLR was introduced that leveraged the SL Graph Convolution Network, GCN [43]. An Indian ISLR was developed using a Deep Learning framework that used CNN for its implementation [44]. An Attention mechanism-based model was developed for gesture recognition to overcome the Complex background challenge [45]. For addressing static and dynamic gestures, a deep learning framework was developed using CNN and Stacked LSTM [46].

Furthermore, much research has been carried out on various regional sign languages. A Sinhala SLR system was introduced for Sinhala word numeric sign recognition [53]. Later the SLR framework was built using syntactically guided recognition of Korean Sign Language [54]. For the Indian Sign Language system, various implementations for Sign Language Recognition were done [55, 56]. For Arabic Sign Language Recognition, a dataset was introduced [57]. Even though there has been a great deal of research in this discipline, there is still a significant research gap that needs to be filled. Most state-of-the-art frameworks have highlighted the importance of integrating manual and non-manual sign gesture recognition. Also, most of the existing work depends on one or the other wearable devices or depth camera sensors for data acquisition, making the system expensive. Such wearable devices also restrict the signer movements, making it unnatural and uncomfortable for the users to perform sign gestures. Other existing challenges in SLR include occlusion, sub-sign reasoning, feature extraction from complex backgrounds, large vocabulary and scalability, signer dependent variations, integration of global and local features, and so on.

## 3 PROPOSED METHODOLOGY

This section focuses on the specifics of implementing the proposed framework. Ultimately, for dynamic gestures, considering a sign representation video, $s = \{s_f\}_{f=1}^{F}$ having total $F$ number of frames, the main objective is to anticipate the respective sign label, $W_L$, and for a given static gesture image $i$ the main objective is to anticipate the respective sign label $I_L$. HNN-ISLR is signer-independent and is implemented in such a way to overcome issues such as varying skin tone, varying illumination conditions, complex background, facial expression, varying hand size, and so on.

### 3.1 Pre-processing of Isolated Sign Language Datasets

The Isolated ISL dataset created consists of both static and dynamic signs. Most of the alphanumeric sign gestures were static, but few were dynamic, such as the sign representation of the English alphabet "J," "Z," and so on. The word sign representation videos were dynamic. Various preprocessing techniques were used on the static sign representation images. The RGB to HSV conversion was done to segregate the intensity and the colour of the images, since HSV is resilient in uncertain lighting and illumination conditions. Gaussian blur was used to remove the random noise from the HSV converted images. Using the Keras image data generator, the preprocessed images were then augmented. Image augmentation techniques such as horizontal flipping, width and height shifting, zooming, rotation, translation, and shearing were used to augment the dataset. Figure 1 illustrates the Pre-processing and data Augmentation process on the static sign images.

In the case of dynamic sign representation videos, the videos were converted into frames for sign recognition. Each video produces a different number of frames based on the duration of the sign action performed. So to handle such variations and bring uniformity in the number of frames

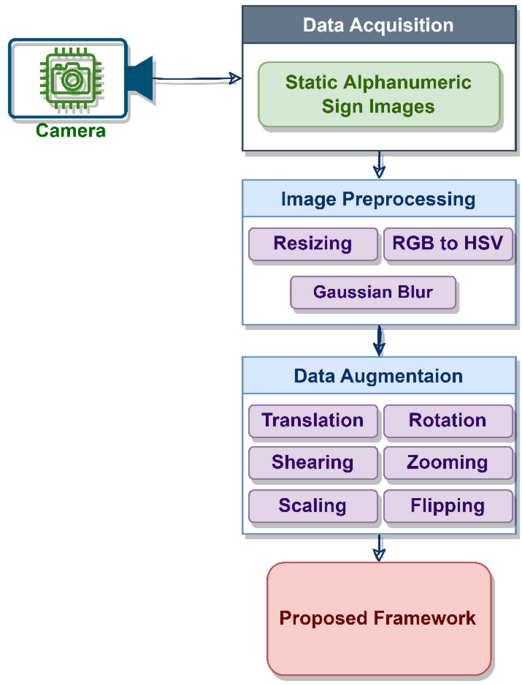


Fig. 1. Preprocessing and data augmentation for static sign representations.

produced by all the videos, we have exploited the sampling method in our framework. The sampling method helps get a predefined number of frames from each video. In sampling the average total number of frames, $Av_f$, in all the videos are calculated and kept as a threshold value for setting up a predefined number of frames to be fed as input to the HNN-ISLR. $Av_f$ can be formulated as given in Equation (1). As given in Equation (1), $T_f$ refers to the overall total number of frames in all videos and $V_n$ refers to the total number of videos:

$$Av_f = \frac{T_f}{V_n}. \tag{1}$$

The frames extracted from the dynamic sign videos and the preprocessed static images were then utilized by the proposed HNN-ISLR Architecture for the SLR task. Here the threshold value selected was $Av_f$, hence from each video, first $Av_f$ number of frames are extracted for the recognition task. When the number of frames in certain videos are less than the $Av_f$ value, the last frame is repeated to increase the frame numbers up to the $Av_f$ value.

### 3.2 Proposed Hybrid Neural Framework for ISLR

The main motive of developing an ISLR system is to recognize the sign representations of isolated words, alphabets, and digits. We propose a Hybrid Neural Network Architecture for the recognition of Isolated Indian and Russian Sign Language Recognition (HNN-ISLR). The proposed HNN-ISLR helps recognize the Indian and Russian SL representations of English words, alphabets, and digits. Figure 2 depicts the framework that has been proposed.

The lack of publicly available Isolated Indian and Russian Sign Language datasets necessitated the need to create and collect a new dataset. We have created our Isolated SL dataset consisting of Indian and Russian sign representations of English Alphabets, digits, and words with the help

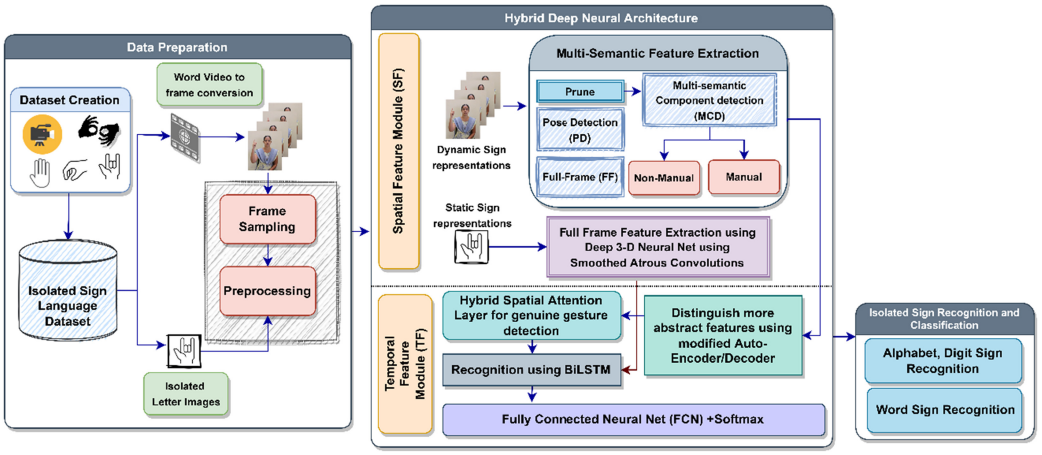


Fig. 2. Proposed HNN-ISLR framework.

of multiple signer volunteers. This novel dataset has been used for experimenting with our proposed framework. After creating the dataset, the sign representations were divided into static and dynamic gestures. The image frames were extracted for dynamic gestures, and Frame Sampling was done to set predefined frame numbers for all the dynamic gestures (videos). As discussed earlier, the dynamic and the static gestures were preprocessed using standard preprocessing techniques. These preprocessed images and frames were then fed as input to the proposed HNN-ISLR framework. The proposed HNN-ISLR framework consists of two main modules: **Spatial Feature Module (SF)** and **Temporal Feature Module (TF)**, contributing to spatial and temporal feature extraction. In the SF module, the multi-semantic, i.e., manual and non-manual features, are extracted from the gestures. The implementation is carried out in two phases, first for static SL gestures and second for dynamic SL gestures.

For static sign representations, the images are fed through the convolutional process of the SF module, and then the feature maps are fed to the TF module. In the case of dynamic sign representations, the frames are fed as input to the SF module, wherein the frames are inputted through three sub-modules, namely, **Multi-semantic Component Detection (MCD), Pose Detection (PD)**, and **Full-Frame Module (FF)**. In the MCD sub-module, the face and hand of the signer are detected and separately pruned. The skeletal pose is detected and extracted from the video frames in the PD sub-module. The full-frame and the complex background are fed to the hybrid neural network in the FF module. As a result of the spatial feature extraction, the output of SF in the form of spatial feature maps generated from the three above-stated sub-modules is then fed to the TF module. The frame spatial feature maps are fed to the auto-encoders in the TF module for the sequential and temporal feature extraction. After the sequential and temporal feature extraction, the extracted features are fed to the modified hybrid Attention Layer. The attention layer will help in the segregation of genuine gestures from non-useful transition sign representations. The segregated genuine gestures will be then fed to the **Bidirectional LSTM (Bi-LSTM)** for further recognition and classification of the gestures. The features are fed to the fully-connected layer with a softmax layer at the output consisting of Recognition of Isolated Letters and Word Sign labels. The detailed proposed architecture is shown in Figure 3.

*3.2.1 Spatial Feature Detection and Extraction.* There are two main modules for dynamic sign recognition SF and TF, as shown in Figure 3. The SF module processes each frame and creates a set of spatial features for face, hand, full-frame, and pose. The SF module can be divided into Phase-I

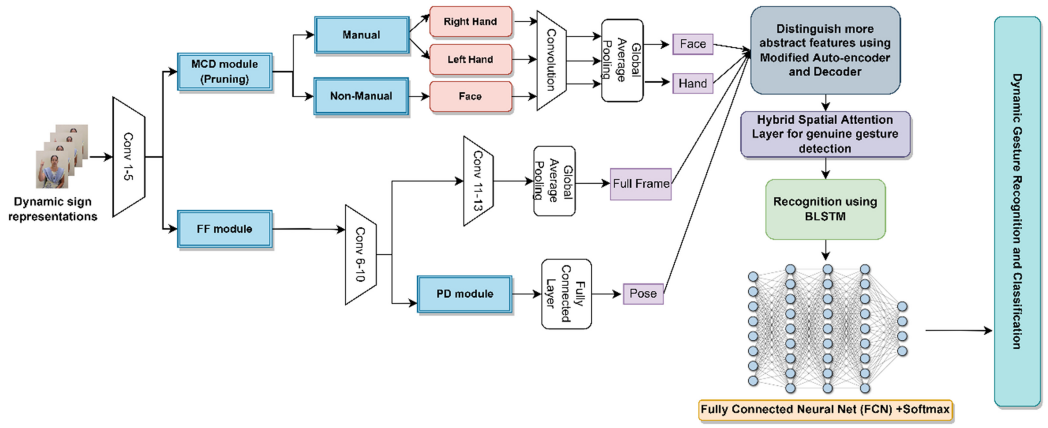


Fig. 3. Detailed architecture of HNN-ISLR framework.

for spatial feature extraction from static sign representations and Phase-II for spatial feature extraction from dynamic sign representations separately. In short, the Phase-I module is developed for feature extraction Static posture, and Phase II is developed for feature extraction from Dynamic gestures. In Phase-I, the static sign images are processed through Deep 3D Neural Net and smoothed Atrous Convolutions (Deep 3D NAC). The 3D Conv nets apply a three-dimensional filter to the dataset that traverses through three directions ($x$, $y$, and $z$) to derive the low-level feature representations. A three-dimensional volume space, including a cube or cuboid, is its output shape. The spacing between the elements in kernels is defined by dilated or Atrous convolution operations. The receptive view of the kernels expands due to the spacing in this sort of convolution; for instance, the field of view obtained by the $3 \times 3$ kernel having a dilation rate of 2 would be the same as a field of view obtained by the $5 \times 5$ kernel. The complexity stays the same; however, distinct features are generated. However, while using atrous convolution, we encounter an issue called Gridding Artifacts wherein the neighbouring output units are derived from a completely different set of input units. The performance of models that use alternative convolution suffers due to this independence. By smoothing the atrous convolution itself, our solutions address the gridding problems. Since our input is two-dimensional image data, the first step is converting each image into 3D shapes along with height, width, and channel information. Since the input is an RGB image, the channel value is 3 (slicing red, green, and blue layers). The 3D Conv Net consists of an input layer with 16, 16, 16, 3 as dimensions and an output layer with ten as dimensions. Four Conv Net layers are applied to have an increased order of size of the filter with a fixed size of kernel ($3 \times 3 \times 3$). After applying convolution, two max-pooling layers are applied after the second and the fourth Conv Net layer. Here, we have used batch normalization and have introduced dense layers along with two layers accompanied by dropout for avoiding overfitting. Here the spatial feature maps are extracted from the static sign representations.

In Phase II of the SF module, the dynamic sign representation frames are passed through the Multi-Semantic Feature Extraction framework. The multi-semantic features here include the feature extraction from the different physical body parts. In Phase II of the SF module, we have used 2D Convolution Neural Net (Conv Net), considering VGG-16 as a backbone network model to generate the multi-semantic features derived from the face, hands, pose, and full-frame as well. This multi-semantic SF framework is divided into three modules discussed earlier: the MCD, FF, and PD. In the MCD module, the manual and non-manual features are detected and extracted. The manual semantic features comprise hand shape, orientation, hand positions, and so on. The non-manual

sematic features comprise mouth shape, facial expression, eye gaze, and so on. In the MCD module, the multi-semantic features are extracted using Grad-CAM and Camshift techniques. To detect the non-manual semantic features, we have generated saliency maps using **Gradient-weighted Class Activation Map (Grad-CAM)**. For tracking the manual semantic features, we have used the Camshift algorithm. After the detection of face and hands using Grad-CAM and Camshift, respectively, the manual and non-manual patches are pruned from the output of the 5th layer of Conv Net ($56 \times 56 \times C_5$). The patch pruning size for the face is considered to be $16 \times 16$, and for both the hands, it is considered to be $24 \times 24$. Each patch's centre point is clamped into a range to ensure that it does not cross the original feature map's border. In the PD module, for detecting and extracting the pose, we have integrated *two* deconvolution layers after the 10th Conv Net layer, which results in the upsampling of the frame to $56 \times 56$. To generate $k$ anticipated heat maps, the output is passed into a point-wise convolutional layer. The position of each heat map's matching keypoint is supposed to exhibit the highest response value. We have selected key points at both the wrists, elbows, shoulders, and nose and hence $k = 7$. A soft-argmax layer has been used to differentiate the key point prediction for sequential learning. So considering heat maps, $\{M_k\}_{K=1}^{k}$, for $k$ heat maps. For $M_K \in \mathbb{R}^{h \times w}$, each $M_k$ is inputted through spatial softmax layer as given in Equation (2), where the value of $M_K$, at a given position $(i,\ j)$ is denoted as $M_{i,j,K}$ and the probability of $K$ at $(i, j)$ is denoted by $P(K)_{i,j}$:

$$P(K)_{i,j} = \frac{e^{M_{i,j,K}}}{\sum_{i=1}^{h}\sum_{j=1}^{w} e^{M_{i,j,K}}}. \tag{2}$$

Over the whole probability map, the anticipated values of coordinates along the $x$-axis and $y$-axis are determined as given in Equation (3):

$$(\hat{x},\ \hat{y})_K = \left( \sum_{i=1}^{h}\sum_{j=1}^{w} \frac{i=1}{h-1} P(K)_{i,j},\ \sum_{i=1}^{h}\sum_{j=1}^{w} \frac{j=1}{w-1} P(K)_{i,j} \right), \tag{3}$$

where the anticipated normalized position of $K$ is $J_K = (\hat{x},\ \hat{y})_K \in [0, 1]$ and $(\widehat{x} \times (h-1)+1,$ $\hat{y} \times (w-1)+1)$ is the respective position of $(\hat{x},\ \hat{y})$ in a feature map having $h \times w$ dimension. In the FF module, the complete frame is passed through all the layers of the Conv Net. The predicted $K$ key points are then converted to a one-dimensional vector by flattening and then finally to extract the pose cues; it is then inputted to a Fully Connected Neural Net (FCN) with ReLU. The convolved feature maps of both the hands and the face are pruned and passed separately through several layers of Conv Net. For both hands, weight-sharing Conv Net layers were exploited as most sign representations depend on coordination among both hands. The output of the MCD, FF, and the PD modules were concatenated along the channel dimensions. Lastly, we employed global average pooling to generate feature vectors from the face, hands, pose, and full-frame feature maps. By inputting the frames $s = \{s_f\}_{f=1}^{F}$ through the SF Module, the feature extraction was done as given in Equation (4).

$$\left\{\{v_{f,z}\}_{z=1}^{Z}, \{J_{f,K}\}_{K=1}^{k}\right\}_{f=1}^{F} = \{\psi_\theta(s_f)\}_{f=1}^{F} \tag{4}$$

where the SF function module is denoted as $\psi_\theta(\cdot)$ and the parameters of the SF module are represented as $\theta$. $J_{f,K} \in \mathbf{R}^2$ is the position of the given $k$, key point at the $f^{th}$ frame. The feature vector of the multi-semantic cues, $z$, at the $f^{th}$ frame is denoted as $v_{f,z} \in \mathbf{R}^{C_z}$. In our work, as we have extracted four visual cues including face, hand, pose and full-frame hence we have set $Z = 4$.

*3.2.2 Temporal Feature Extraction.* The TF module learns and extracts the temporal and sequential features from the spatial feature maps. The spatial features extracted from the visual cues

using SF modules are then passed to the autoencoders at the TF module. We have used the **Variational Auto-Encoders (VAE)** in our framework. The VAE is an autoencoder with regularised training to combat overfitting and ensure that the latent space has desirable properties that allow for generative processes. VAE encodes an input as distribution throughout the latent space rather than a single point, thereby introducing some regularisation of the latent space. Using VAE, more abstract features have been extracted, such as palm, palm radius, hand orientation, fingertips, and so on. The rather intuitive loss function of VAEs, comprising of a reconstruction term and a regularisation term, can be meticulously determined, employing, in particular, the statistical approach of variational inference, given a basic underlying probabilistic model to represent the features. The last layer of the VAE has been eliminated as our main objective is to extract abstract features from the spatial feature maps. When training a VAE, the loss function that is minimized is made up of a reconstruction term that attempts to make the encoding-decoding configuration as efficient as possible, and a regularisation term, on the latent layer, that tries to regularise the organization of the latent space by making the encoder's distributions close to a standard normal distribution. The regularisation term is the Kulback-Leibler divergence between the returned distribution and a typical Gaussian. The proposed framework deploys an activation function, namely, **Gaussian Error Linear Unit (GELU)** and the VAE. The GELU outperforms the ReLU activation function. ReLU activation has allowed Neural Network models to converge faster and better than sigmoid functions, whereas dropout generalizes the framework by multiplying several activations by 0 at random. Both of these approaches are used to determine a neuron's output. Despite this, the two work independently of one another. GELU aspires to bring them together. In addition, Zoneout, a new regularizer, stochastically multiplies the input by 1. By combining these three functions by stochastically multiplying the input by 1 or 0 and deterministically, we can obtain the output of the activation function. The neuron input $n$ by $Bernoulli(\phi(n))$ denoted by $m$, hence we have the value of $\phi(n)$ as shown in Equation (5):

$$\phi(n) = P(X \leq n) \text{ and } X \sim \mathcal{N}(0,1). \tag{5}$$

Since neural input follows a normal distribution, notably after Batch Normalization, any activation function's output must be deterministic rather than stochastic. Hence, the transformation's expected value is to be found as given in Equation (6):

$$\mathrm{E}[\mathrm{mn}] = n\mathrm{E}[m]. \tag{6}$$

As m is the *Bernoulli Random Variable* the Expected value is given in terms of $\phi(n)$. Hence, Equation (6) could be rewritten as Equation (7):

$$\mathrm{E}[\mathrm{mn}] = n\phi(n). \tag{7}$$

As $\phi(n)$ is the Gaussian Cumulative Distribution as well as it is mostly formulated along with the error function, the GELU activation function is defined as given in Equation (8):

$$GELU(n) = nP(X \leq n) = n\phi(n). \tag{8}$$

Using this modified VAE, more abstract high-level features have been extracted. After the sequential and temporal feature extraction, the extracted features are fed to the hybrid Attention Layer, which incorporates LightBGM with a Hybrid Attention Layer. The objective of the attention mechanism is to offer context and guide the decoder network's attention to a specific range of encoder outputs. The primary objective of attention mechanisms is to generate a weighted summary of the source sequence to help in decoding. The context vector is a term used to describe this summary. Considering $\mathcal{H} = [h_1, h_2, h_3, \ldots, h_{z+1}]$ as the hidden states, $\alpha_v$ as the attention weights, $o_v$ as the

output at each hidden state, and $c_v$ as the context vector, generally, $c_v$ can be formulated as shown in Equation (10):

$$\alpha_v = softmax\,(o_v W \mathcal{H}^{\intercal}), \tag{9}$$

$$c_v = [\alpha_v \mathcal{H}_v]. \tag{10}$$

The attention weights for different cues would be the same in this case. Hence to differentiate the weights for different cues, the Attention Layer proposed here is taken as a combination of **Segmentation (SeA)** and **Spatial Attention (SpA)** Mechanism incorporated in parallel.

The SeA mechanism assigns attention weights to each feature extracted separately. The value of the context vector $c_v$, using SeA, can be formulated as shown in Equation (13), where $\alpha_v$ is segmented uniformly across $z + 1$ channel parts:

$$o_v = [o_{v,1},\ o_{v,2}, \ldots, o_{v,z+1} \tag{11}$$

$$\alpha_v = softmax\left(o_{v,z} W_z \mathcal{H}_z^{\intercal}\right), \tag{12}$$

$$c_v = [\alpha_{v,\ 1},\ h_1,\ \alpha_{v,\ 2},\ h_2, \ldots, \alpha_{v,\ z+1},\ h_{z+1}]. \tag{13}$$

The SpA leverages the inter-spatial interdependence of features to generate a spatial attention map. The spatial attention, which is complementary to the channel attention, focuses on where there could be an informative component, as opposed to the channel attention, which focuses on where there is a channel. SpA initially uses average-pooling and max-pooling operations along the channel vector and concatenates those to create an effective feature descriptor to quantify spatial attention. A convolution layer is applied to the concatenated feature descriptor to build a spatial attention map $M_s(F) \in \mathbb{R}^{h \times w}$. $M_s$ helps to encode, where to suppress or emphasize. The feature map channel information is aggregated by using two pooling operations that generate two 2D maps, which are $F_s^{avg} \in \mathbb{R}^{1 \times h \times w}$ and $F_s^{max} \in \mathbb{R}^{1 \times h \times w}$, denoting average pooling and max-pooling features, respectively. The 2D spatial attention map is created by concatenating and convolving these with a regular convolution layer. The SpA is computed as shown in Equation (15). where the sigmoid function is denoted as $\sigma$, and the convolution with $7 \times 7$ filter size is denoted by $C^{7\times7}$. The output must take the shape of a 1D linear vector. Direct inputs in the form of cubic or rectangular shapes are not possible. Hence, the set of feature maps is then flattened into 1D attention weight vectors. The attention weights of SeA and SpA are concatenated for formulating the attention weights:

$$M_v\,(F) = \sigma\left(C^{7\times7}\left([Averagepool\,(F); Maxpool\,(F)]\right)\right), \tag{14}$$

$$M_v\,(F) = \sigma\left(C^{7\times7}\left(\left[F_s^{avg}, F_s^{max}\right]\right)\right). \tag{15}$$

After extracting genuine and non-useful gestures using the Hybrid Attention Mechanisms, the output is passed through the BiLSTM to recognize and classify the sign representations. LSTM helps to learn the sequential and temporal features of genuine gestures and, hence, helps recognize the corresponding sign representations. The LSTM consists of three gates, namely, **input-gate (*i*), output-gate (*o*)**, and **forget-gate (*g*)**. The *g*-gate is used to decide which information has to be discarded and which information has to be retained in the state cell. The *i*-gate selects the values to be updated. A *tanh* function ($e_t$) is leveraged for generating a vector for the new election values. The *o*-gate decides the output part. The LSTM functions are briefly formulated in Equations (16)–(21):

$$g_t = \sigma\left(W_g\,[h_{t-1},\ n_t] + b_g\right), \tag{16}$$

$$i_t = \sigma(W_i\,[h_{t-1}, n_t] + b_i), \tag{17}$$

$$\tilde{e}_t = \tanh\,(W_e\,[h_{t-1}, n_t] + b_e), \tag{18}$$

$$e_t = g_t * e_{t-1} + i_t * \tilde{e}_t, \tag{19}$$

$$o_t = \sigma(W_o [h_{t-1}, n_t] + b_o), \tag{20}$$

$$h_t = o_t * \tanh(e_t). \tag{21}$$

Since LSTM only absorbs input from previous sequences, the basic LSTM has some restrictions in its performance. However, accessing both past and future information improves accuracy in the sign recognition problem. As it captures both past and future information, **bi-directional LSTM (BiLSTM)** is a promising solution to this problem. Two LSTM layers are trained on the input sequences in bi-directional LSTM to provide additional information from the input and provide quick results. The first LSTM used forward temporal information to train on the input sequence, whereas the second used reverse temporal information to train on the reversed copy of the input sequence. The forward and backward outputs will be concatenated to model the sequence's bi-directional dependency. The effectiveness of BiLSTM is improved as the network depth is extended. In our model, a BiLSTMNet is constructed by stacking three BiLSTM layers with a varying number of hidden units. Additionally, BiLSTM layers have been substituted with LSTM layers to construct the LSTMNet. These networks are followed by a **Fully Connected Neural Net (FCN)** and Softmax Layer for sequence recognition.

## 4 RESULT AND DISCUSSION

### 4.1 Dataset

There are still many challenges in the ISLR, but the major drawback is the lack of publicly available datasets. Even though few datasets are available, they have still not considered many constraints such as multiple signers, large vocabulary, the light intensity of the image, camera distance, camera angle, multiple and single-handed gestures, skin tone, background, and so on. We have considered major features such as multiple signers, skin tone, varying hand size, complex background conditions, static and dynamic gestures, and single and double-handed gestures. We have collected and created both static and dynamic sign representation datasets for ISLR. For Indian sign representation, we have created two isolated sign dataset collections, namely, ISLAN [5] and ISLW. The ISLAN dataset comprises of Indian Sign representation of English alphabets and digits. It is a collection of 700 sign images and 24 videos with single-handed and double-handed gestures. The images and videos were captured with the help of 6 signer volunteers. Individuals with varying skin tones and hand sizes were considered to capture this dataset. The images/videos were captured in an indoor space under normal lighting conditions. Figure 4 shows the Static Indian sign representation of English Alphabets of a single signer. Figure 5 shows static single-handed sign representation of alphabet "A," and Figure 6 shows the irregular double-handed sign representation of the English alphabet "A."

The ISLW dataset consists of Indian Sign representation of English words. ISLW word videos were captured with the help of 7 different signer volunteers with a complex background and varying illumination conditions. It consists of 500 unique word signs with at least five repetitions for each signer. Hence the ISLW dataset consists of a total of 2,500 word videos. The videos have an image resolution of 1,080 p with a frame rate of 30–45 fps. The video files are in MP4 format, and the image frames are in JPEG format. The videos were captured in an indoor space. Figure 7 illustrates the dynamic sign gesture frame extraction for the word "Famous."

The proposed work also involves creating and collecting the Russian Sign Language Dataset. Dataset has been created with having 1,100 unique signs and 37,775 general Russian sign representation. Dataset consisting of 1,100 different Russian Word Signs have been collected and created. The Russian Signs were captured from three different native signer volunteers having five

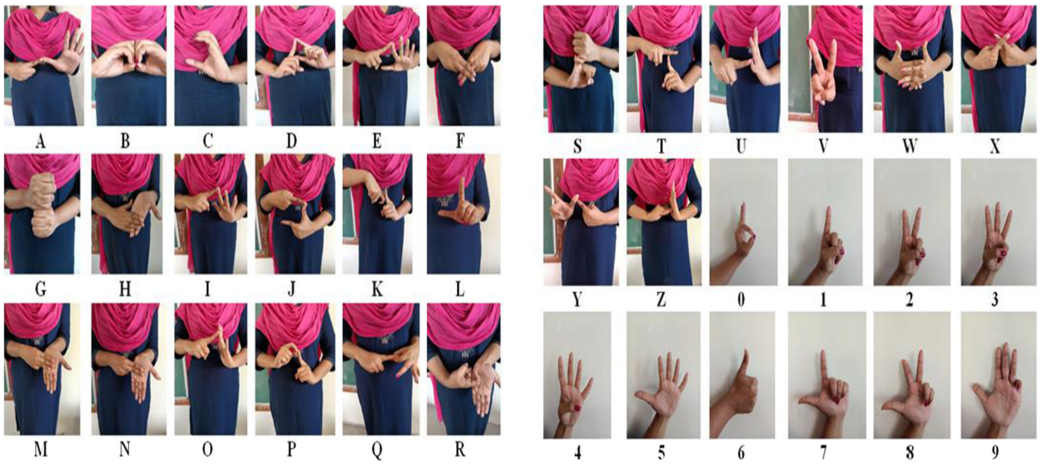


Fig. 4. Static Indian sign representation of English Alphabets of a single signer.

Fig. 5. Static double-handed sign representation of English alphabet "A."

repetitions each. The dataset has been segregated, keeping in mind the words, movement types, and hand shapes. Dataset was then processed to distinguish five types of movements: idle, beginning (from idle to sign), sign, ending (from sign to idle, transitional). Dataset of about 800 different handshapes for only static signs with about seven different signers, in 12,595 video files, each about 3–5 s in duration that is 30 frames multiplied by 3 s that turns out to have 377,850 frames in total. There were different resolutions from 320*240 to 1,920*1,080 and different FPS from 25 to 60 s per frame considered. Uniform backgrounds, but mostly green, were used. Among the word signs, about 30% are one-handed, 30% are symmetric two-handed, 30% are asymmetric two-handed, about 5–10% are combined. The video was also converted to skeletal data via Mediapipe. The errors in the dataset are about 5%. Figure 8 illustrates the Russian Sign Representation of the word "Scissors" by various signers.

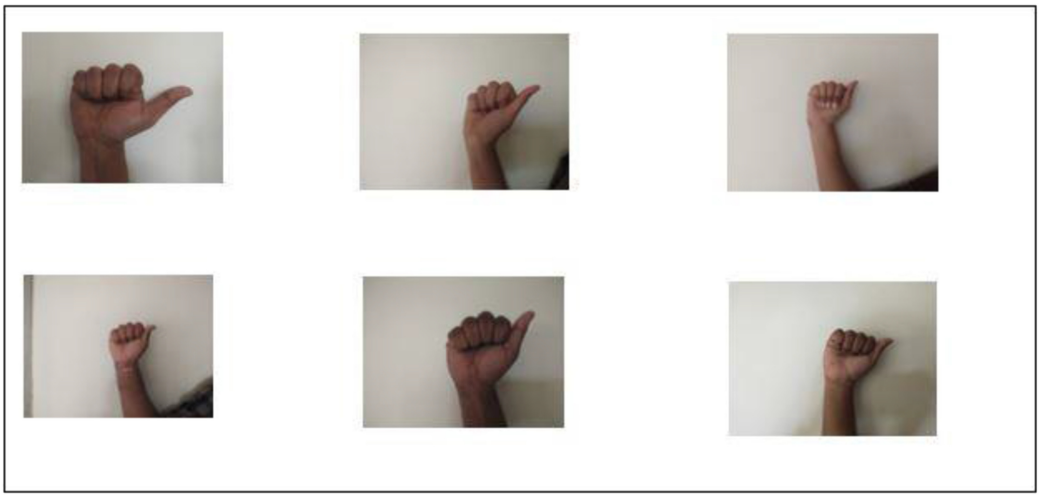

Fig. 6. Static single-handed sign representation of alphabet "A."

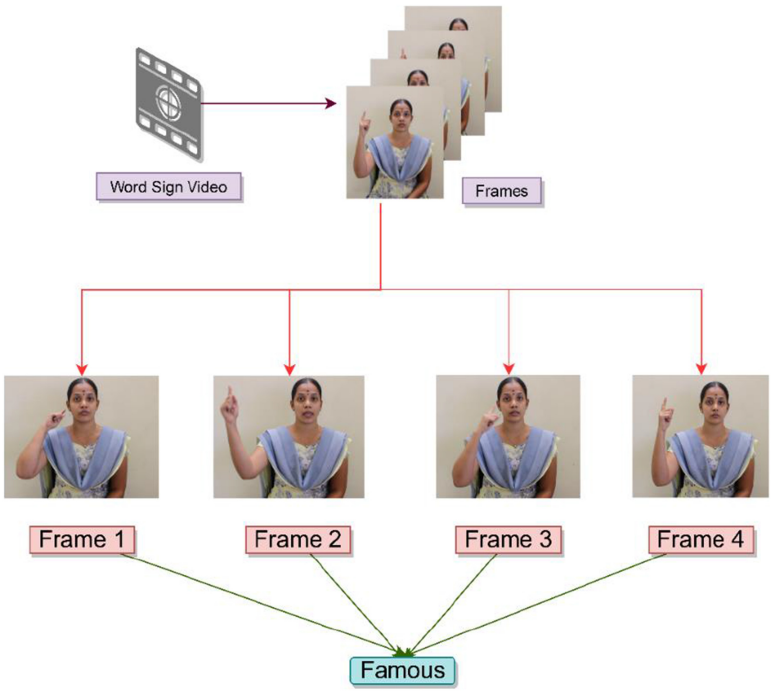


Fig. 7. Dynamic sign gesture frame extraction for the word "Famous."

### 4.2 Result Analysis for the Proposed HNN-ISLR Framework

The proposed HNN-ISLR model was deployed for the newly created and collected Russian and Indian Static and Dynamic Isolated sign Language representation. The results from the proposed model proved to be efficient and have good accuracy. This section will discuss the results obtained from the proposed HNN-ISLR model. After preprocessing the datasets, the static alphabet sign

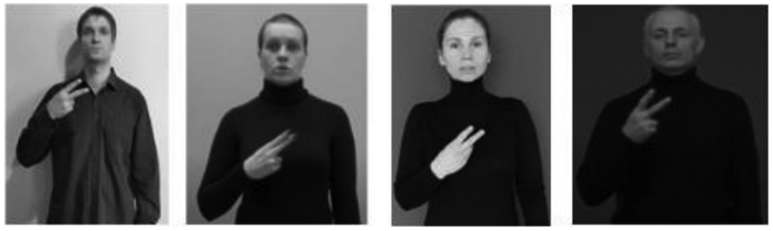

Fig. 8. Multi-signer sign representation of word “Scissors” [47].

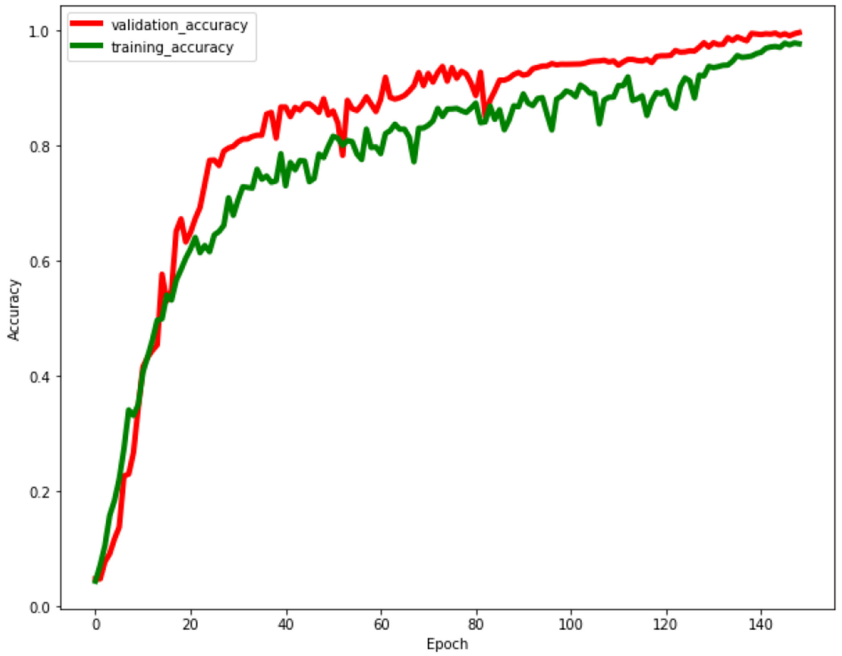


Fig. 9. Result accuracy for static sign representation recognition.

images were sent to the Phase-I of SF module and the word sign video frames were sent to the Phase II of the SF module. The spatial feature extraction for static images and the dynamic frames was done independently using separate modules. After extracting the spatial features for temporal feature extraction, the feature maps of the dynamic frames and static images were inputted to the TF module, followed by recognizing static postures and dynamic gestures. For the loss function, the Categorical Cross-Entropy Loss function was used. For efficient feature extraction and recognition, the proposed model uses cross-entropy for fine-tuning the BiLSTM, and L1-loss has been smoothed by using SGD as an optimizer. The proposed HNN-ISLR has experimented with RTX 3,060 GPU using python TensorFlow. The framework was trained for 140 epochs. The model uses early stopping to stop training as the accuracy stops to improve. The experimental results were plotted in the graph as shown in this section.

Figure 9 illustrates the accuracy graph obtained from training and testing the Indo-Russian Static Sign Language Dataset using 140 epochs. The validation accuracy obtained was 99.76%, while the training accuracy obtained was 98.71%. Figure 10 illustrates the loss graph obtained from training and testing the Indo-Russian Static Sign Language Dataset using 140 epochs. Figure 11 illustrates

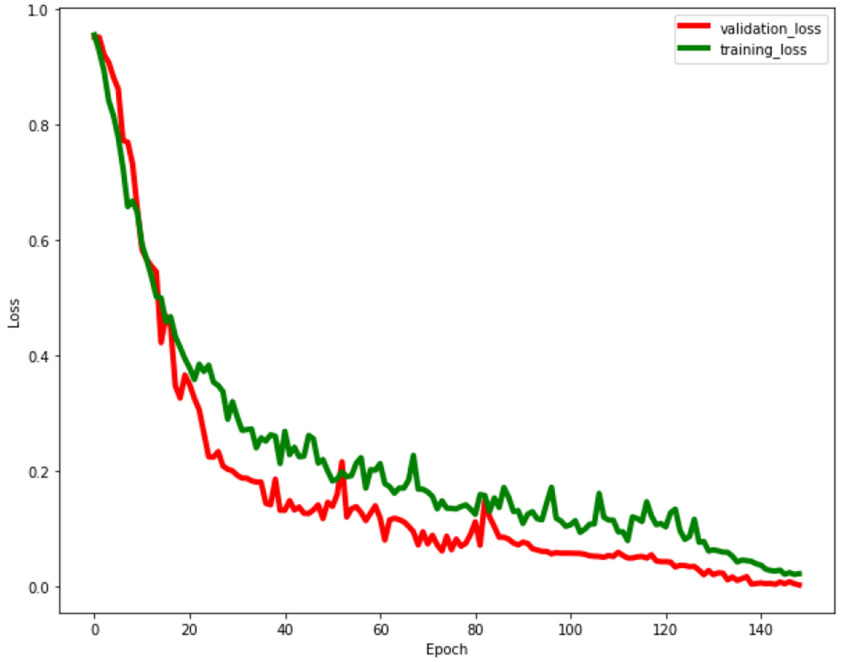


Fig. 10. Result loss for static sign representation recognition.

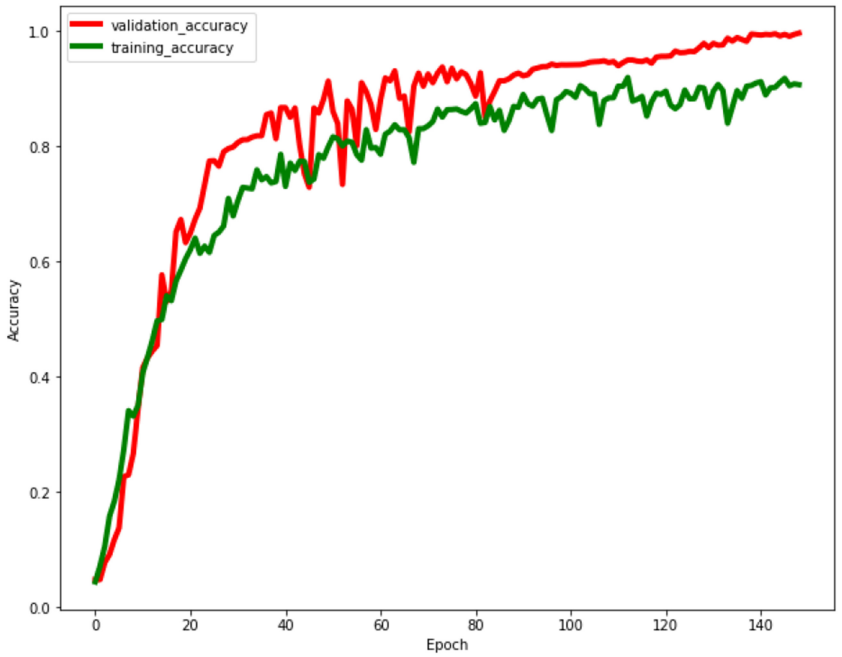


Fig. 11. Result accuracy for dynamic sign representation recognition.

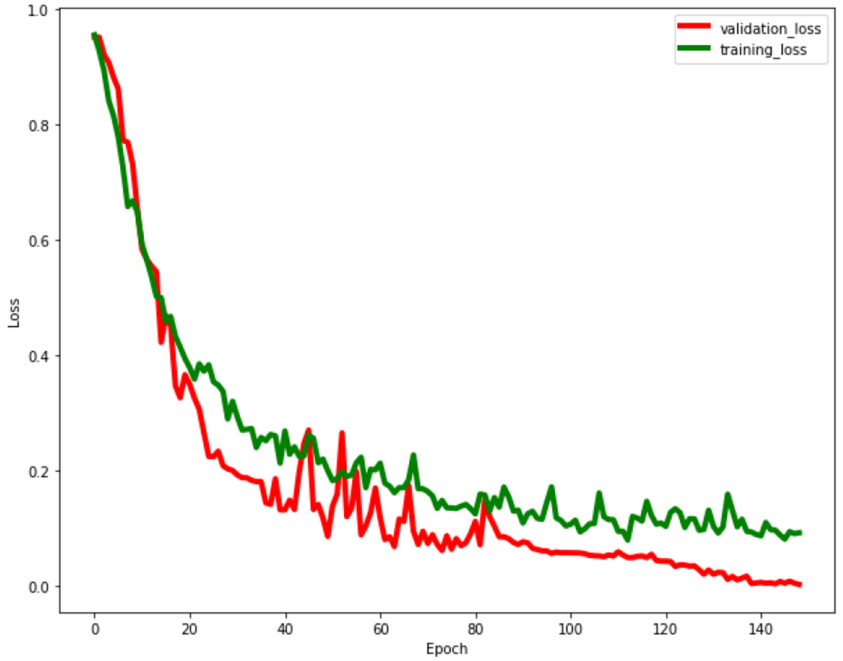


Fig. 12. Result loss for dynamic sign representation recognition.

the accuracy graph obtained from training and testing the Indo-Russian Dynamic Sign Language Dataset using 140 epochs. The validation accuracy obtained was 99.85%, while the training accuracy obtained was 97.71%. Figure 12 illustrates the loss graph obtained from training and testing the Indo-Russian Dynamic Sign Language Dataset using 140 epochs.

From the plotted accuracy graphs of the Static Sign representations and the Dynamic Sign representations, it is evident that the gap between the training and the validation curve is low, indicating that there are very few chances of high overfitting and the model tends to converge faster. From the plotted accuracy graphs of the Static Sign representations and the Dynamic Sign representations, it is evident that the model has a good learning rate.

### 4.3 Result Comparison for Proposed HNN-ISLR with Benchmark Dataset

To further assess the performance of our proposed HNN-ISLR framework with a multilingual SL dataset, we have performed experimentation on two of the benchmark datasets, namely, ArASL [48] and WLASL [49]. ArASL has 32 Arabic static sign classes and nearly 54,000 images. [1] has done experimentation on ArASL with fine-tuned VGG16 and fine-tuned ResNET152 framework, and we have compared our results with this work, and our results proved to be better in performance. The comparison of our proposed method with other datasets helped evaluate the model's performance concerning other Sign Languages, which proved to help build a unified framework for identifying and recognizing sign gestures in multiple Sign Languages. Figure 13 shows the result comparison of static posture recognition of the proposed HNN-ISLR with the baseline model. WLASL is a collection of American word-level sign gesture datasets. WLASL is divided into subsets, including WLAS100, WLASL300, WLASL1000, and WLASL2000. When the classes are ranked in order of the number of videos for each category for each collection, these subsets consist of the top-N classes, wherein N = {100, 300, 1,000, 2,000}. We have analyzed the performance of the proposed model with scalability. The result is plotted as a graph. Figure 14 shows the model's

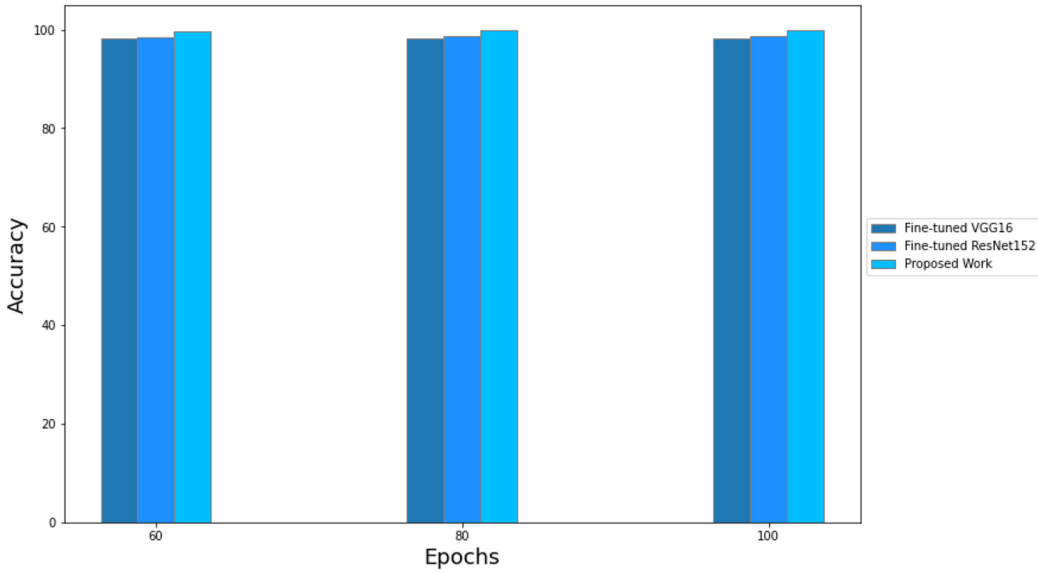


Fig. 13. Result comparison of proposed HNN-ISLR with baseline models for ArASL dataset.

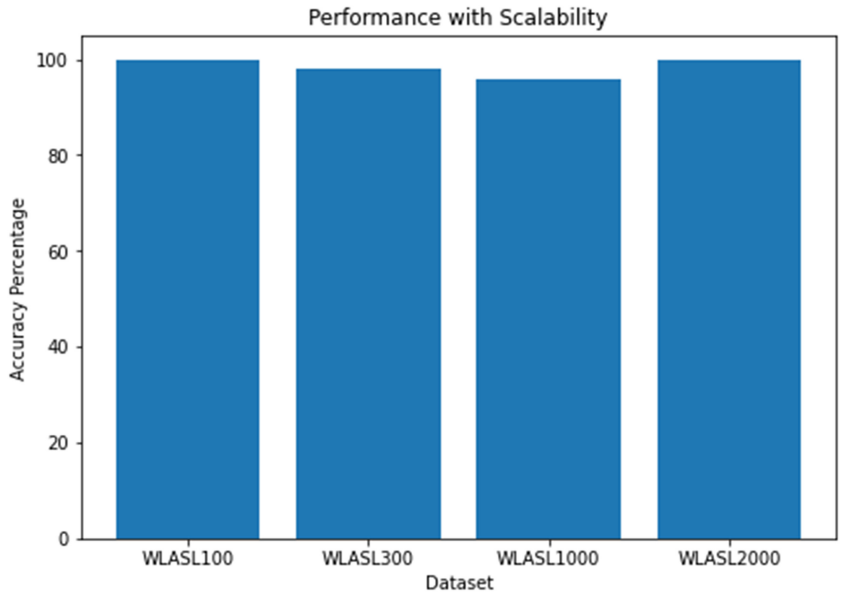


Fig. 14. Performance of proposed work based on Scalability.

performance with an increase in the dataset from WLASL100 to the WLASL2000 dataset. Based on the graph, we can interpret that the performance remains consistent even with the increase in the dataset. We have analyzed compared the performance of our proposed HNN-ISLR architecture on WLASL with other baseline models, which include I3D [49], Pose-GRU [49], Pose-TGCN [49], GCN-BERT [50], Fusion 3 [51], and Multi-Stream [52]. Figure 15 illustrates the performance

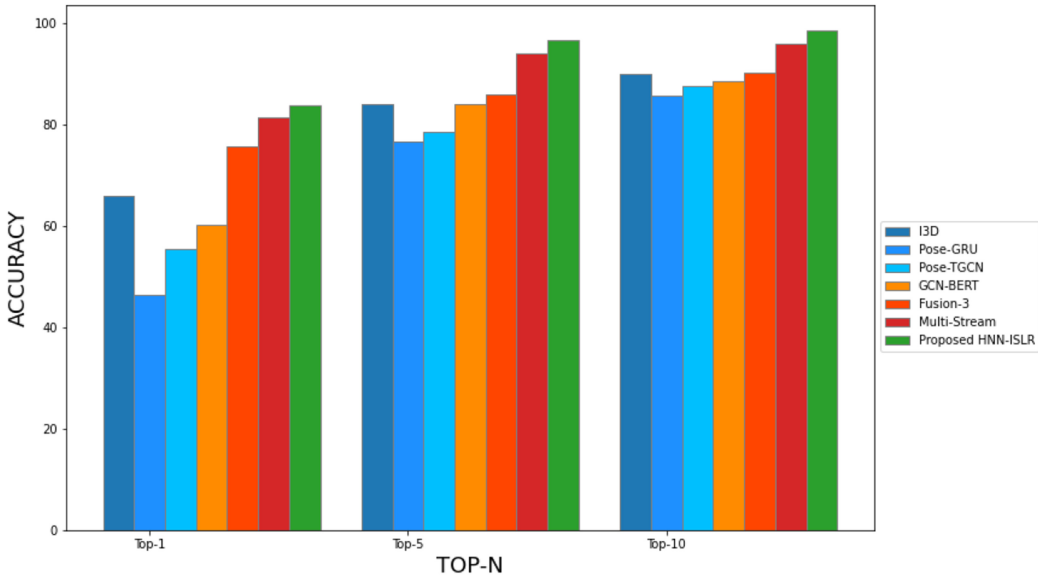


Fig. 15. Result comparison of proposed HNN-ISLR with baseline models for WLASL100 dataset.

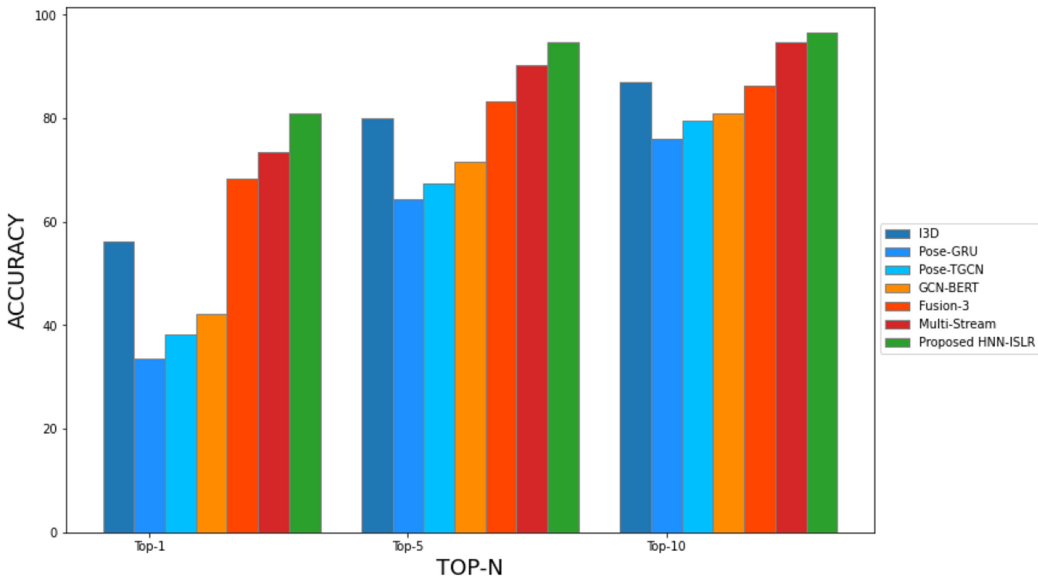


Fig. 16. Result comparison of proposed HNN-ISLR with baseline models for WLASL300 dataset.

comparison of the proposed HNN-ISLR with other state-of-the-art models, for the WLASL100 subset dataset, in terms of accuracy. Figure 16 illustrates the performance comparison concerning accuracy for the WLASL300 subset dataset.

Figure 17 illustrates the performance comparison concerning accuracy for the WLASL1000 subset dataset. Figure 18 illustrates the performance comparison concerning accuracy for the WLASL2000 subset dataset. Compared to other baseline models, our proposed work yields better

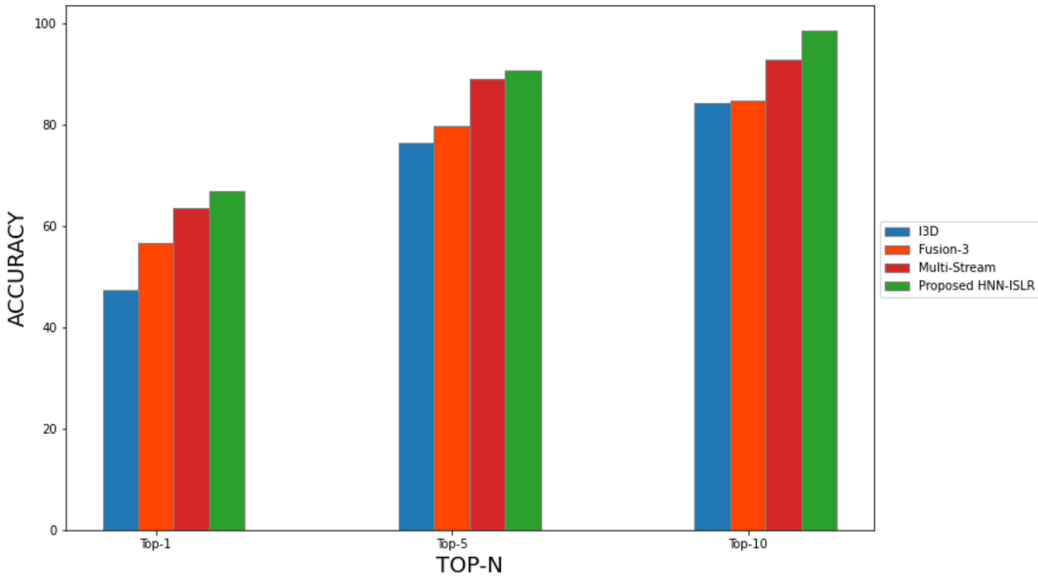


Fig. 17. Result comparison of proposed HNN-ISLR with baseline models for WLASL1000 dataset.

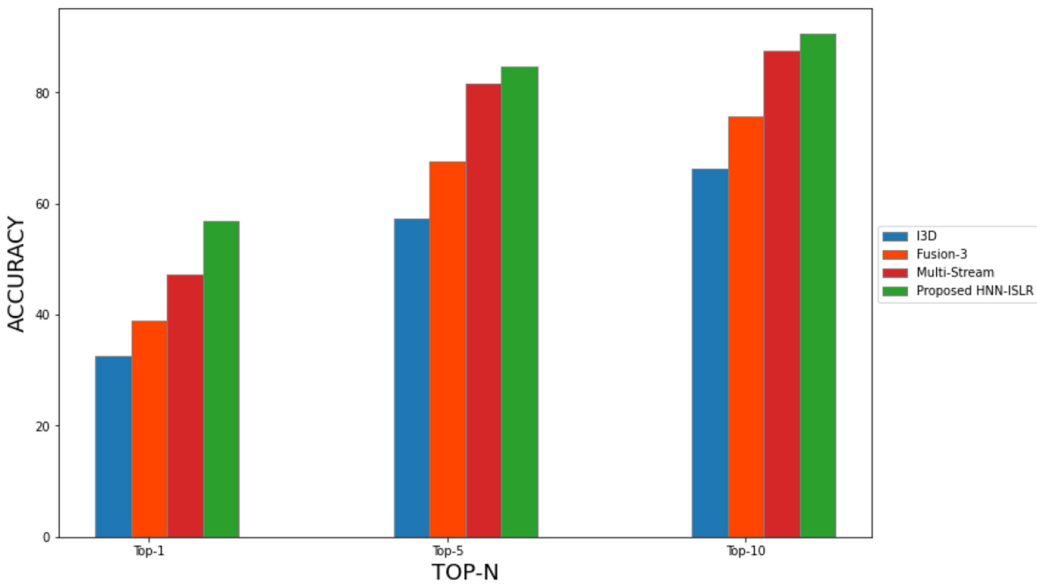


Fig. 18. Result comparison of proposed HNN-ISLR with baseline models for WLASL2000 dataset.

accuracy results. Though, as the number of the dataset increases, our proposed work proves to be performing better and adapt to scalability. As a result, our proposed HNN-ISLR framework is scalable enough and adaptive for Multilingual SLR. Moreover, the ArASL and WLASL datasets are multi-signer datasets. Hence, the graphical representations show that the performance of proposed HNN-ISLR is better than other state of the art methodologies for multilingual, multisigner atatic and dynamic datasets.

## 5 CONCLUSION AND FUTURE WORK

Sign Language is communication for the mute and hard of hearing population. It is very complex to discern and interpret for the rest of the population. SLR system tends to bridge this communication barrier by translating the sign gestures into readable textual form. This research has proposed a novel Hybrid Neural Network approach for recognizing Isolated Indian and Russian Sign language. The proposed architecture is signer-friendly as the signers were not constrained to any clothing restrictions or use any wearable sensors encouraging the natural way of signing. None of the wearable or depth camera sensors was used to develop the framework, thus making the recognition system cheap and affordable. The proposed architecture comprises two main modules, one for spatial semantic feature detection and another for temporal and sequential feature extraction. The spatial semantic feature module is divided into two phases: Phase I generates spatial feature maps for Static SL representations; Phase II generates feature maps for Dynamic SL representations and manual and non-manual feature extraction. The temporal and feature module helps extract more abstract features and decide the genuine and non-useful transitional sign representations. We have created and collected our multi-signer sign language dataset. Our proposed work is adaptive for a signer-variant dataset captured with a complex background, varying illumination conditions, varying skin tone, and varying hand size conditions. The main object of the proposed framework is to detect and classify the Word-level and Alphanumeric-level Sign Language gestures. Our work yields 99.76% accuracy for recognizing Isolated Static signs and 99.85% for recognizing Isolated Dynamic signs. We compared our proposed HNN-ISLR framework's performance with other baseline algorithms using benchmark multilingual sign datasets. We conclude that our proposed work yields better accuracy than other base models based on the performance evaluation. As part of our future work, we would like to build an SLR model for addressing the occlusion issues, sub-sign issues, and a large vocabulary. We would also try to increase our Isolated word-level dataset to cover all the spoken words. We also intend to extend our work in Indian and Russian Continuous Sign Language collection and make it available publicly for further research. We want to build Continuous SLR for the Indian and Russian datasets created.

## REFERENCES


[1] Y. Saleh and G. Issa. 2020. Arabic sign language recognition through deep neural networks fine-tuning. https://www.learntechlib.org/p/217934/.

[2] K. Wangchuk, K. Wangchuk, and P. Riyamongkol. 2020. Bhutanese sign language hand-shaped alphabets and digits detection and recognition (doctoral dissertation, naresuan university). http://nuir.lib.nu.ac.th/dspace/handle/123456789/2491.

[3] X. Jiang, M. Lu, and S. H. Wang. 2020. An eight-layer convolutional neural network with stochastic pooling, batch normalization and dropout for fingerspelling recognition of chinese sign language. *Multimedia Tools Appl.* 79, 21 (2020), 15697–15715.

[4] O. Sevli and N. Kemaloğlu. 2020. Turkish sign language digits classification with CNN using different optimizers. *Int. Adv. Res. Eng. J.* 4, 3 (2020), 200–207.

[5] R. Elakkiya and E. Rajalakshmi Islan. Mendeley Data, Vol. 1. https://data.mendeley.com/datasets/rc349j45m5/1.

[6] S. C. Ong and S. Ranganath. 2005. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 6 (2005), 873–891.

[7] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef, and C. Vogler. 2019. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* 16–31.

[8] R. Cui, H. Liu, and C. Zhang. 2019. A deep neural framework for continuous sign language recognition by iterative training. *IEEE Trans. Multimedia* 21, 7 (2019), 1880–1891.

[9] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li. 2018. Video-based sign language recognition without temporal segmentation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence.*

[10] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 9 (2019), 2306–2320.

[11] H. Wang, X. Chai, and X. Chen. 2019. A novel sign language recognition framework using hierarchical grassmann covariance matrix. *IEEE Trans. Multimedia* 21, 11 (2019), 2806–2814.
[12] J. Pu, W. Zhou, and H. Li. 2019. Iterative alignment network for continuous sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4165–4174.
[13] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden. 2020. Sign language transformers: Joint end-to-end sign language recognition and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10023–10033.
[14] O. Koller, J. Forster, and H. Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Comput. Vision Image Understand.* 141, (2015) 108–125.
[15] D. Hendrycks and K. Gimpel. 2016. Gaussian error linear units (gelus). Retrieved from https://arXiv:1606.08415.
[16] W. Sandler. 2012. The phonological organization of sign languages. *Lang. Ling. Compass* 6, 3 (2012), 162–182.
[17] R. Elakkiya. 2021. Machine learning-based sign language recognition: A review and its research frontier. *J. Ambient Intell. Human. Comput.* 12, 7 (2021), 7205–7224.
[18] S. Diwakar and A. Basu. 2008. A multilingual multimedia Indian sign language dictionary tool. In *Proceedings of the International Joint Conference on Natural language Processing (IJCNLP'08)*. 57.
[19] S. K. Liddell and R. E. Johnson. 1989. American sign language: The phonological base. *Sign Lang. Studies* 64, 1 (1989), 195–277.
[20] P. Eccarius and D. Brentari. 2007. Symmetry and dominance: A cross-linguistic study of signs and classifier constructions. *Lingua* 117, 7 (2007), 1169–1201.
[21] Z. Ren, J. Yuan, J. Meng, and Z. Zhang. 2013. Robust part-based hand gesture recognition using kinect sensor. *IEEE Trans. Multimedia* 15, 5 (2013), 1110–1120.
[22] C. Wang, Z. Liu, and S. C. Chan. 2014. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE Trans. Multimedia* 17, 1 (2014), 29–39.
[23] Z. Liu, X. Chai, Z. Liu, and X. Chen. 2017. Continuous gesture recognition with hand-oriented spatiotemporal feature. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 3056–3064.
[24] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz. 2016. Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4207–4215.
[25] A. Kumar, K. Thankachan, and M. M. Dominic. 2016. Sign language recognition. In *Proceedings of the 3rd International Conference on Recent Advances in Information Technology (RAIT'16)*. IEEE, 422–428.
[26] O. Aran, I. Ari, L. Akarun, B. Sankur, A. Benoit, A. Caplier, P. Campr, and A. H. Carrillo. 2009. Signtutor: An interactive system for sign language tutoring. *IEEE MultiMedia* 16, 1 (2009), 81–93.
[27] I. N. Sandjaja and N. Marcos. 2009. Sign language number recognition. In *Proceedings of the 5th International Joint Conference on INC, IMS, and IDC*. IEEE, 1503–1508.
[28] Q. Yang. 2010. Chinese sign language recognition based on video sequence appearance modeling. In *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications*. IEEE, 1537–1542.
[29] S. Hore, S. Chatterjee, V. Santhi, N. Dey, A. S. Ashour, V. E. Balas, and F. Shi. 2017. Indian sign language recognition using optimized neural networks. In *Information Technology and Intelligent Transportation Systems*. Springer, Cham, 553–563.
[30] R. Agarwal. 2021. Bayesian k-nearest neighbour based redundancy removal and hand gesture recognition in isolated indian sign language without materials support. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*. IOP Publishing, 012126.
[31] S. Reshna and M. Jayaraju. 2017. Spotting and recognition of hand gesture for Indian sign language recognition system with skin segmentation and SVM. In *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET'17)*. IEEE, 386–390.
[32] A. Chaudhary and J. L. Raheja. 2018. Light invariant real-time robust hand gesture recognition. *Optik* 159 (2018), 283–294.
[33] K. Shenoy, T. Dastane, V. Rao, and D. Vyavaharkar. 2018. Real-time indian sign language (ISL) recognition. In *Proceedings of the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT'18)*. IEEE, 1–9.
[34] A. Tyagi and S. Bansal. 2021. Feature extraction technique for vision-based indian sign language recognition system: A review. *Comput. Methods Data Eng.* 39–53.
[35] R. Agarwal. 2021. Bayesian K-nearest neighbour based redundancy removal and hand gesture recognition in isolated indian sign language without materials support. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*. IOP Publishing, 1116, 1 (2021), 012126.
[36] M. Mukushev, A. Sabyrov, A. Imashev, K. Koishibay, V. Kimmelman, and A. Sandygulova. 2020. Evaluation of manual and non-manual components for sign language recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association (ELRA'20)*.

[37] A. Tunga, S. V. Nuthalapati, and J. P. Wachs. 2021. Pose-based sign language recognition using GCN and BERT. In *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'21)*. 31–40.
[38] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden. 2018. Neural sign language translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7784–7793.
[39] K. M. Lim, A. W. C. Tan, C. P. Lee, and S. C. Tan. 2019. Isolated sign language recognition using convolutional neural network hand modelling and hand energy image. *Multimedia Tools Appl.* 78, 14 (2019), 19917–19944.
[40] S. Aly and W. Aly. 2020. DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access* 8, 83199–83212.
[41] D. Li, X. Yu, C. Xu, L. Petersson, and H. Li. 2020. Transferring cross-domain knowledge for video sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6205–6214.
[42] R. Rastgoo, K. Kiani, and S. Escalera. 2020. Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools Appl.* 79, 22965–22987.
[43] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu. 2021. Skeleton aware multi-modal sign language recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3413–3423.
[44] S. Sharma and S. Singh. 2021. Recognition of Indian sign language (ISL) using deep learning model. *Wireless Personal Commun.* (2021), 1–22.
[45] G. Jianchun, G. Jiannuan, and W. Lili. 2021. Gesture recognition method based on attention mechanism for complex background. *J. Phys.: Conf. Ser.* 1873, 1 (2021), 012009.
[46] O. Mazhar, S. Ramdani, and A. Cherubini. 2021. A deep learning framework for recognizing both static and dynamic gestures. *Sensors* 21, 6 (2021), 2227.
[47] M. G. Grif, R. Elakkiya, A. L. Prikhodko, M. A. Bakaev, and E. Rajalakshmi. 2021, Raspoznavanie recognition of Russian and Indian sign languages based on machine learning. *Analysis and Data Processing Systems* 3, 83 (2021), 53–74.
[48] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf. 2019. ArASL: Arabic alphabets sign language dataset. *Data Brief* 23 (2019), 103777.
[49] D. Li, C. Rodriguez, X. Yu, and H. Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1459–1469.
[50] A. Tunga, S. V. Nuthalapati, and J. P. Wachs. 2021. Pose-based sign language recognition using GCN and BERT. In *Proceedings of the IEE Workshop on Applications of Computer Vision (WACV'21)*. 31–40.
[51] A. A. Hosain, P. S. Santhalingam, P. Pathak, H. Rangwala, and J. Kosecka. 2021. Hand pose guided 3d pooling for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3429–3439.
[52] M. Maruyama, S. Ghose, K. Inoue, P. P. Roy, M. Iwamura, and M. Yoshioka. 2021. Word-level sign language recognition with multi-stream neural networks focusing on local regions. Retrieved from https://arXiv:2106.15989.
[53] M. Punchimudiyanse and R. G. N. Meegama. 2017. Animation of fingerspelled words and number signs of the sinhala sign language. *ACM Trans. Asian Low-Res. Lang. Info. Process.* 16, 4 (2017), 1–26.
[54] H. Y. Jung, J. H. Lee, E. Min, and S. H. Na. 2019. Word reordering for translation into korean sign language using syntactically-guided classification. *ACM Trans. Asian Low-Res. Lang. Info. Process.* 19, 2 (2019), 1–20.
[55] P. Kumar and S. Kaur. 2020. Sign language generation system based on indian sign language grammar. *ACM Trans. Asian Low-Res. Lang. Info. Process.* 19, 4 (2020), 1–26.
[56] J. Singha and K. Das. 2013. Recognition of indian sign language in live video. Retrieved from https://arXiv:1306.1301.
[57] A. A. I. Sidig, H. Luqman, S. Mahmoud, and M. Mohandes. 2021. KArSL: Arabic sign language database. *ACM Trans. Asian Low-Res. Lang. Info. Process.* 20, 1 (2021), 1–19.
[58] K. K. Verma and B. M. Singh. 2021. Deep multi-model fusion for human activity recognition using evolutionary algorithms. *Int. J. Interact. Multimedia Artific. Intell.* 7 (2021), 44–58.
[59] K. K. Verma, B. M. Singh, H. L. Mandoria, and P. Chauhan. 2020. Two-stage human activity recognition using 2D-ConvNet. *Int. J. Interact. Multimedia Artific. Intell.* 6 (2020), 11.
[60] M. Boháček and M. Hrúz. 2022. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 182–191.
[61] S. Srivastava, A. Gangwar, R. Mishra, and S. Singh. 2022. Sign language recognition system using tensorflow object detection API. Retrieved from https://arXiv:2201.01486.

Received 17 January 2022; revised 23 March 2022; accepted 8 April 2022