# INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition

Advaith Sridhar
advaithsridhar08@gmail.com
IIT Madras
Chennai, Tamil Nadu

Rohith Gandhi Ganesan
rgg296@nyu.edu
IIT Madras
Chennai, Tamil Nadu

Pratyush Kumar
pratyush@ai4bharat.org
IIT Madras
Chennai, Tamil Nadu

Mitesh Khapra
mitesh@ai4bharat.org
IIT Madras
Chennai, Tamil Nadu

## ABSTRACT

Indian Sign Language (ISL) is a complete language with its own grammar, syntax, vocabulary and several unique linguistic attributes. It is used by over 5 million deaf people in India. Currently, there is no publicly available dataset on ISL to evaluate Sign Language Recognition (SLR) approaches. In this work, we present the Indian Lexicon Sign Language Dataset - INCLUDE - an ISL dataset that contains 0.27 million frames across 4,287 videos over 263 word signs from 15 different word categories. INCLUDE is recorded with the help of experienced signers to provide close resemblance to natural conditions. A subset of 50 word signs is chosen across word categories to define INCLUDE-50 for rapid evaluation of SLR methods with hyperparameter tuning. As the first large scale study of SLR on ISL, we evaluate several deep neural networks combining different methods for augmentation, feature extraction, encoding and decoding. The best performing model achieves an accuracy of 94.5% on the INCLUDE-50 dataset and 85.6% on the INCLUDE dataset. This model uses a pre-trained feature extractor and encoder and only trains a decoder. We further explore generalisation by fine-tuning the decoder for an American Sign Language dataset. On the ASLLVD with 48 classes, our model has an accuracy of 92.1%; improving on existing results and providing an efficient method to support SLR for multiple languages.

## KEYWORDS

Indian Sign Language; INCLUDE; BiLSTM; XGboost; ASLLVD

**ACM Reference Format:**
Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. 2020. INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition. In Proceedings of the 28th ACM International Conference on Multimedia (MM'20), October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413528

## 1 INTRODUCTION

The amount of video content being created and consumed continues to rise sharply [63]. This has led to a continued focus on using deep learning for tasks on video data such as video classification [30], object detection [42, 53], object tracking [62], action recognition [9], visual question answering [20] and video encoding [43]. Many of these tasks have practical applications, while some have important accessibility applications as well. In particular, sign language recognition (SLR) can be thought of as an action recognition task.

Sign languages are systems of visual communication, used primarily by people from deaf communities around the world. In its most common form, words and phrases are signed by gesturing with fingers, hands, arms and facial expressions. Sign languages are fully developed languages with their own grammar and lexicon. Further, they differ from region to region and are often not mutually intelligible with each other [17], though some languages do possess similarities. They also differ from the spoken languages of a given area in terms of lexicon and rate of articulation [4].

Indian Sign Language (ISL) is a complete language with its own grammar, syntax, vocabulary and other linguistic attributes [17]. ISL varies considerably from its western counterparts in a variety of ways, most noticeably its lexicon, which displays a high level of iconicity [17]. While most other sign languages have a few compound signs (signs consisting of two or more signs), in ISL the compounding system is pervasive [17]. As an example, the word *Brother* is signed by compounding signs for *Male* and *Sibling*, and the word *Wife* is signed by compounding signs for *Female* and *Marry*. This is a deviation from the spoken languages of India as well, which are known for their array of complex kinship terms. The range of such compositions possible allows for a particularly large lexicon in ISL. Another aspect in which ISL differs from other sign languages is in the space around the body used for signing. In ISL, the upper signing space signifies distance and authority, which is in marked contrast to many European and North American Sign Languages where distance is shown through a horizontal plane in front of the signer [17]. These and other deviations of ISL from other sign languages warrant development and evaluation of new methods for Sign Language Recognition for ISL. Further, the large

*MM '20, October 12–16, 2020, Seattle, WA, USA.*

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413528

number of deaf people in India – over 5 million – highlights the large practical impact of any deployable solution.

Development and evaluation of machine learning models critically depend on the existence of resources in the form of large and standardised datasets. There have been a few earlier efforts in creating datasets for ISL. For example, [31, 51, 60] have created ISL video datasets, while [29, 52, 54] have created image datasets with a single image denoting a sign. Crucially, none of these datasets are publicly available, disallowing any further study on them. These datasets also have two other major limitations. First, the number of classes, i.e., number of distinct word signs, is significantly low. The largest number of classes is found in [31], which has 10 different signer videos per sign, for 80 ISL signs. These numbers fall significantly short of a moderate sized vocabulary for ISL. Second, the videos are often artificially constrained. [51, 52, 54] have the images cropped so that only the hands are visible in their datasets. [31, 52, 54, 60] use pictures or videos with uniform backgrounds, often with the colour of the signer's clothes matching the background colour. [51] and [29] use additional equipment such as Kinect cameras and data gloves, respectively. These constraints on the dataset limit the applicability of models trained on them in real-world scenarios. Thus, currently, no dataset for ISL is available at the size and quality expected for machine learning research.

Across multiple sign languages, Sign Language Recognition (SLR) has been an active area of research over the last two decades [6, 8, 14, 18, 22–24, 32, 35, 38, 39, 57, 61]. In SLR, the correct word (class) must be predicted by observing the sequence of video frames belonging to a particular sign. Most signs in sign language are of extremely short duration (1-3 seconds in length). The signs vary from each other only in terms of subtle finger, hand, arm and facial feature movements, resulting in low inter-class variance. Furthermore, different signers have their own individual styles of enacting signs, which results in signs having large intra-class variance. Thus, SLR is a challenging task; we review the prior work and results for different sign languages in the next section.

In this work, we aim to address the lack of public datasets and scalable models for ISL recognition. We make three contributions:

- We propose the INCLUDE - Indian Lexicon Sign Language Dataset - for ISL. It is the first publicly available ISL dataset and compares favorably with datasets on other sign languages: it contains 0.27 million frames with 263 classes from 15 different word categories. The videos in INCLUDE are recorded with the help of the deaf community. The video background, resolution and lighting are chosen to provide close resemblance to real-world scenarios. To enable rapid evaluation of deep learning models on the dataset, we propose a smaller subset, named INCLUDE-50, with 50 classes.
- We analyse multiple deep learning pipelines for SLR on the INCLUDE dataset. These pipelines include choices for augmenting the dataset, extracting features with pre-trained networks, and using different backbones for encoding and decoding. Amongst different pipelines, we find the combination of feature extraction with a pre-trained pose detection network, encoding with a pre-trained MobileNet network, and decoding with trained bidirectional LSTMs has an accuracy of 94.5% on INCLUDE-50 and 85.6% on INCLUDE. This is the first such systematic study of models for SLR on ISL.
- In our identified deep learning pipeline, the only learnable parameters are in the bilinear LSTM decoders. This motivates the analysis of fine-tuning the decoder for other sign languages. We evaluate this with a subset of the ASLLVD dataset [1] with 48 classes, where a class is chosen if it has at least 8 example videos. Our pipeline achieves a high accuracy of 92.1%. This compares favourably with the best reported result of 91.58% on a similar sized subset with a fully trained 3DCNN. Thus, our method demonstrates that with a common core and a multiple fine-tuned decoders we can support recognition on multiple sign languages.

INCLUDE and all models presented in this paper are open-sourced.

The rest of the paper is organised as follows. In Section 2 we summarise the related work by listing datasets and proposed approaches for SLR. We then detail the collection procedure and data format of the INCLUDE dataset in Section 3. In Section 4, we describe different augmentation methods and deep learning models. We evaluate the models and summarise results on INCLUDE and other datasets in Section 5.

## 2 RELATED WORK

In this section, we first describe different sign language datasets and then summarise different methods for sign language recognition.

### 2.1 Datasets on Sign Language

Several datasets have been proposed for sign languages across the world. The ASLLVD dataset [1] is a well studied dataset on the American Sign Language lexicon. It has over 3,300 distinct signs, each produced by 1-6 native signers. However, only 48 out of the 3,300 signs in the dataset have 8 or more videos. All of the videos of the dataset are recorded with uniform background for easy segmentation of hands and face. Another dataset for ASL Lexicon Recognition is MS-ASL [28]. The MS-ASL dataset has 1,000 classes of signs which are signed by over 222 signers. This provides larger diversity than the ASLLVD dataset. The RWTH-Boston-50 [44] is another 50 class ASL lexicon dataset that is fairly popular. RWTH-PHOENIX-Weather 2014 [33] is a German sign language dataset aimed at continuous sign language recognition. The dataset has over a million frames and a vocabulary size of 1,081 unique words. The dataset is recorded from a public television weather broadcast and has sentences performed by 9 unique signers. Ko et al. [32] propose a dataset for continuous Korean Sign Language recognition called KETI, with over 14,000 videos and 100 sentences.

As discussed in the introduction, some datasets have been proposed for the Indian Sign Language. Rekha et al. [52] uses a dataset for ISL containing 290 static images across 26 letters of the alphabet which can be used for finger-spelling. These static images are cropped to ensure only the hands of the signers are visible, making it easier to extract features. Nandy et al. [46] proposes a dataset with around 600 videos across 22 classes. The videos are gray-scale and are cropped to ensure only the hands are visible. Kishore et al. [31] proposes a dataset that has a considerable vocabulary of videos (800 videos across 80 classes) but have constraints imposed on how the videos are recorded. Thus, existing datasets for ISL do not have the quality and size required to train machine learning models. Further, none of the datasets on ISL are publicly available.

## 2.2 Sign Language Recognition Methods

Sign Language Recognition is a well-established research problem. In standard or isolated SLR, single videos correspond to a single sign, while in continuous SLR, a single video comprises of multiple signs. Continuous SLR is the harder problem requiring both segmentation and classification of the video. In this section, we will only focus on isolated SLR as that is the task for the INCLUDE dataset.

*Dependence on Additional Instrumentation:* In early research, special methods such as data-gloves were used to collect high quality low-dimensional data to feed into smaller machine learning models [10, 19, 27, 55]. For instance, Kim et al. [27] use data-gloves to detect the motion of hands and fingers of the signer and a fuzzy min-max neural network to classify the signs, their proposed model can identify 25 Korean Signs with an accuracy of 85%. Starner et al. [56] use a desk-mounted or wearable camera that captures video input and HMMs to perform the recognition task. The proposed HMM achieves 92% and 98% accuracy for desk-mounted and wearable camera positions respectively on a 40 word lexicon of American Sign Language (ASL). Yang et al. [45] extract trajectories from videos and use a time-delayed neural network to classify signs. They report an accuracy of 99% on 40 ASL Gestures. Clearly, these methods do not apply for datasets like INCLUDE which do not have any instrumentation as is expected in a real-world deployment.

*Using 3D or Depth Sensors:* More recent methods use the camera output to recognise the signs. This recognition is more efficient when utilising the depth information from the cameras to segment the hand [38] [39] [61]. The additional depth information allowed classical machine learning models such as Support Vector Machines [57] and Random Forests [8] to better recognize signs, with the SVM achieving an accuracy of 86% across 73 classes of American Sign Language. Kumar et al. [37] classifies 50 Indian Sign Language signs using a HMM, with an accuracy of 95%. Cooper et al. [13] uses 3D tracking data to classify 984 signs with an accuracy of 71.4%. These set of results also do not apply to the INCLUDE dataset as we do not record depth information in the video.

*Using Deep Neural Networks:* Advancements in learning based methods allowed training of Convolutional Neural Networks (CNNs) and 3D-Convolutional Neural Networks (3D-CNNs) to classify signs [3, 21, 22, 26, 34, 48, 49]. He et al. [22] uses a 3D-CNN to identify American Sign Language from videos. The proposed model was able to achieve an accuracy of 90% on a sub-sample of 50 labels from the ASLLVD dataset. Jing et al. [26] also uses 3DCNNs on American Sign Language, achieving accuracies of 92.88% on a 100 classes. Pigou et al. [48] uses a CNN on 20 words from Italian Sign Language and achieves an accuracy of 91.7%. Koller et al. [34] uses a CNN with an iterative EM algorithm to label signs from Danish Sign Language, achieving an accuracy of 50.6% across 60 classes. Pigou et al. [49] uses a CNN with residual connections on a 100 signs each from Dutch and Flemish sign language, to achieve a top-10 accuracy of 73.5% and 56.4% respectively. Bantupalli et al. [3] uses the Inception network on a picture American Sign Language dataset to achieve an accuracy of 91% across 150 classes.

In this paper, we evaluate a similar approach using deep CNNs.

*Use of Feature Extractors:* One approach to SLR is to break each sign into smaller units (sub-units), which are then used as features for classification [50, 58, 59]. Han et. al [58] further goes on to use a boosting algorithm, AdaBoost, on weak base learners such as decision trees and HMMs. A more recent approach has been to use features extracted from networks designed to recognise human body pose in a single frame and optical flow vectors across frames. Zanchettin et al. [15] extracts skeletal pose points from a subset of 20 signs of ASLLVD and uses a graph CNN to classify the points, achieving an accuracy of 56.82%. Charles et al. [11] and Ko et al. [32] extract skeletal key points on continuous sign language, with Ko et al. achieving an accuracy of 55.28% on 105 Korean Sign Language sentences. Konstantinidis et al. [36] extract skeletal points from 64 signs used in Argentinian Sign Language and establish an accuracy of 98% using an LSTM on the same.

These features are significantly smaller than the input images and therefore require smaller machine learning models. In this paper, we take a similar approach to extract features using the OpenPose network and then process the reduced feature set.

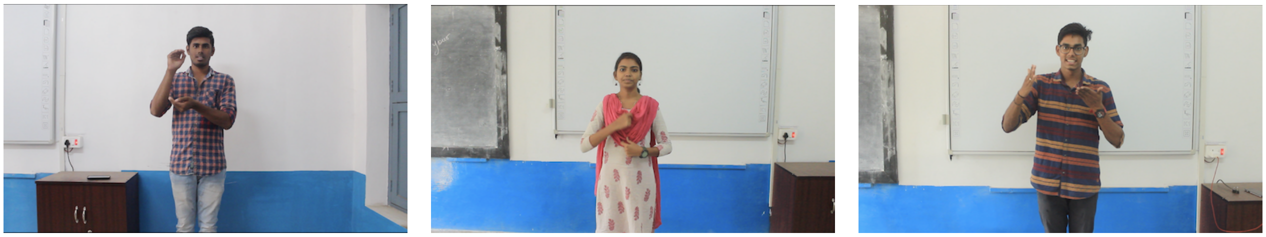
*Results on ASLLVD:.* We separately summarise the main results reported on the ASLLVD, which is based on the American Sign Language. The ASLLVD has over 9,800 videos covering over 3,300 ASL signs. Many of these signs have very few videos; indeed only 48 signs have 8 or more videos. Given this size, many researchers have reported results on chosen subsets of ASLLVD. Theodorakis et al. [59] select a subset of 97 signs from ASLLVD and report an accuracy of 63.15%, using the sub-units approach. Lim et al. [41] calculate histograms of optical flow features on a subset of 20 signs from ASLLVD, reporting an accuracy of 85%. Zanchettin et al. [15] also pick a subset of 20 signs, using a graph CNN to obtain an accuracy of 56.82%. Mao et al. [22] combine videos from multiple views with 3DCNNs, to achieve an accuracy of 91.58% on a 50 sign subset of ASLLVD. There have also been attempts to add additional information to the ASLLVD corpus to aid classification. Bilge et al. [5] add text descriptions to each class of ASLLVD and attempt zero-shot learning, while Lim et al. [40] train their models using the RWTH-Boston-50 dataset and test on a subset of the ASLLVD.

# 3 THE INCLUDE DATASET

The main contribution of this paper is the Indian Lexicon Sign Language Dataset, or INCLUDE. Our principles in the design of INCLUDE are two-fold:

- The videos should resemble real-life scenarios.
- The dataset should cover a diverse set of signs and provide multiple videos for each sign.

Based on these principles, the following procedure was established to collect the data. The dataset was created by recording at a school for the deaf with the help of 7 senior students. This was done by following all administrative procedures and with the official support of the school. All 7 students were experienced signers in ISL having completed several years of education in the same.



(a) Baby

(b) Suit

(c) Clean

**Figure 1: Snapshots from the INCLUDE dataset for various classes. The subjects are experienced signers. No constraints were applied on the background or clothing.**

Each class or sign in the dataset is signed by multiple signers, with each signer signing between 2 to 6 videos per class.

Subjects were made to stand facing the camera at a distance of 5-7 feet from it. Each video is such that the plane of signing is completely captured, which ranges from hip height to an arms length above the shoulder. Videos were shot in bright, natural lighting with no effort made to regulate the signer's clothes or signing style. The background varies across videos, though all videos have been shot in a classroom setting with clutter such as desks, boards and cabinets. Each video is labelled with its corresponding lexeme. It begins at the start of the sign and ends once the sign is completed. Three sample shots from different sign videos are shown in Figure 1.

We now discuss the statistics of the dataset. INCLUDE comprises of 4,287 videos across 263 classes. Each class corresponds to a single sign in ISL. These classes belong to 15 broad categories, covering popular sets of words in ISL. The categories are: *Adjectives, Animals, Clothes, Colours, Days and Time, Electronics, Greetings, Means of Transport, Objects at Home, Occupations, People, Places, Pronouns, Seasons, Society*. A complete list of words in the dataset is given in the Appendix. The category-wise statistics are summarised in Table 2. A majority of the signs are 2-4 seconds in length, with an average duration of 2.57 seconds. Each video is of resolution 1920x1080, with a frame rate of 25 fps. In total, the INCLUDE dataset has 0.27 million frames covering 263 different signs.

**Table 1: INCLUDE: Size of each category**

| Category | Number of Classes | Number of Videos |
|---|---|---|
| Adjectives | 59 | 791 |
| Animals | 8 | 166 |
| Clothes | 10 | 198 |
| Colours | 11 | 222 |
| Days and Time | 22 | 306 |
| Electronics | 10 | 140 |
| Greetings | 9 | 185 |
| Means of Transport | 9 | 186 |
| Objects at Home | 27 | 379 |
| Occupations | 16 | 225 |
| People | 26 | 513 |
| Places | 19 | 399 |
| Pronouns | 8 | 168 |
| Seasons | 6 | 85 |
| Society | 23 | 324 |
| **Total** | **263** | **4287** |

### 3.1 INCLUDE-50

Training deep learning models on a large video dataset can be computationally expensive. To enable fast evaluation of different models, we propose a smaller subset called INCLUDE-50 with 50 of the 263 classes. The classes chosen are: *Bank, Bird, Black, Boy, Brother, Car, Cell phone, Court, Cow, Death, Dog, Election, Fall, Fan, Father, Girl, Good Morning, Hat, Hello, House, I, Monday, Paint, Pen, Priest, Red, Shoes, Shop, Summer, T-Shirt, Teacher, Thank you, Time, White, Window, Year, Large, Dry, Good, Happy, Hot, It, Long, Loud, New, Quiet, Short, Small, Train Ticket, You (plural)*. The classes cover all 15 categories and have been chosen on the basis of frequency of use. In total, INCLUDE-50 contains 958 videos and 60897 frames, and is slightly more than one-fifth of the dataset.

For both INCLUDE and INCLUDE-50 we identify a random stratified subset of 20% of the videos as the test or validation sets. The rest of the videos are the train sets. In the later sections, all experimental results report the accuracy on the respective test sets of models trained on the respective train sets.

**Table 2: Key statistics of the two proposed datasets**

| Characteristic | INCLUDE-50 | INCLUDE |
|---|---|---|
| Categories | 15 | 15 |
| Words | 50 | 263 |
| Videos | 958 | 4287 |
| Avg Videos per Class | 19.16 | 16.3 |
| Avg Video Length | 2.54s | 2.57s |
| Min Video Length | 1.44s | 1.28s |
| Max Video Length | 6.16s | 6.16s |
| Frame Rate | 25fps | |
| Resolution | 1920x1080 | |

The dataset will be made publicly available as part of the final paper. We are also developing a web tool which would allow crowdsourcing of additional videos to add to this dataset.

## 4 METHODS FOR SLR ON INCLUDE

In this section, we discuss different methods for SLR on INCLUDE. We combine different choices for augmentation, feature extraction, encoding and decoding to create different methods. We begin with a discussion on multiple approaches for augmentation. Then, we discuss the deep learning pipeline which consists of *feature extractor*, *encoder* and *decoder*. The results of evaluating the methods on INCLUDE-50 and INCLUDE are presented in the next section.

### 4.1 Augmentation

Augmentation is the process of creating variants of the input data while preserving naturalness, such that trained models generalise better. We perform the following augmentations on the input videos:

- Center Crop - crop to only the center of the video. Based on experiments, we choose to crop 40% of the image in a frame.
- Horizontal Flip - Flipping the frames of the video along the vertical axis, i.e., a signer signing with the right hand will appear to sign with the left hand after flipping.
- Up-sample - Duplicate frames to increase the number of frames in a video. Based on experiments, we uniformly sample and replicate 50% of the frames in a video. The resulting video has 1.5x frames compared to the original video.
- Down-sample - Uniformly select and drop frames from the video. Based on experiments, we uniformly sample 35% of the frames and drop them. The resulting video has 0.65x frames compared to the original video.

Thus, a single video is passed through these augmentations to obtain different versions. As we discuss in the experimental results in the next section, augmenting the dataset leads to improved accuracy of models on the INCLUDE-50 dataset.

### 4.2 Feature Extraction

Video is high-dimensional data: A single sign of 3 seconds is represented by over 150 million integers. In the INCLUDE-50 dataset, we need to map such high-dimensional data to one of 50 signs using just tens of example videos per sign. To enable such learning, we first extract lower-dimensional features using pre-trained models.

OpenPose [7] is a model for real-time multi-person pose estimation on single two-dimensional images. Rather than top-down object detection, it uses a bottom-up non-parametric representation, called Part Affinity Fields (PAFs). A PAF is a two-dimensional vector field that encodes the direction from one part to another of a specific limb, for all such limbs in the image. For instance, the PAF of the right forearm would contain vectors denoting the direction between right shoulder and right elbow in the regions identified as right forearms and zero elsewhere. In addition, OpenPose estimates about 135 key-points[1] in each image which detect different body parts such as faces, hands, and feet. Thus, given an image, OpenPose computes the key-points and the PAF.

By passing an input video through OpenPose frame-wise, we can thus compute three sets of features:

- *Key-points vector*: A vector of x- and y-coordinates of each of the key-points for each frame.
- *Pose video*: A frame-wise pixel map of all limbs.
- *PAF video*: A frame-wise pixel map with PAF aggregations.

The key-points vector is a low-dimensional representation with at most 96 numbers denoting each frame. The Pose and PAF videos are the same resolution as the input image, but are significantly sparse. On average for the INCLUDE-50 dataset, we found that the number of pixels zeroed out in a frame of Pose video is 96.5% and the corresponding number of a PAF video is 79.7%. Thus, with a pre-trained OpenPose tool we can extract features in the form of the key-point vector, pose video, and the PAF video.

Different combinations of these features are used in different methods. For instances, instead of sending the original video, we can send either the pose or PAF video or a concatenation of both to the deep learning pipelines. We discuss these choices in the remainder of this section.

### 4.3 Deep Learning Pipelines

Based on the features extracted using the OpenPose network, we study two broad methods for SLR on the INCLUDE-50 dataset. In the first method we use only the key-points vector and an efficient XGBoost classifier. In the second method, we use combinations of the Pose and PAF videos with different encoder and decoder deep neural networks (DNNs).

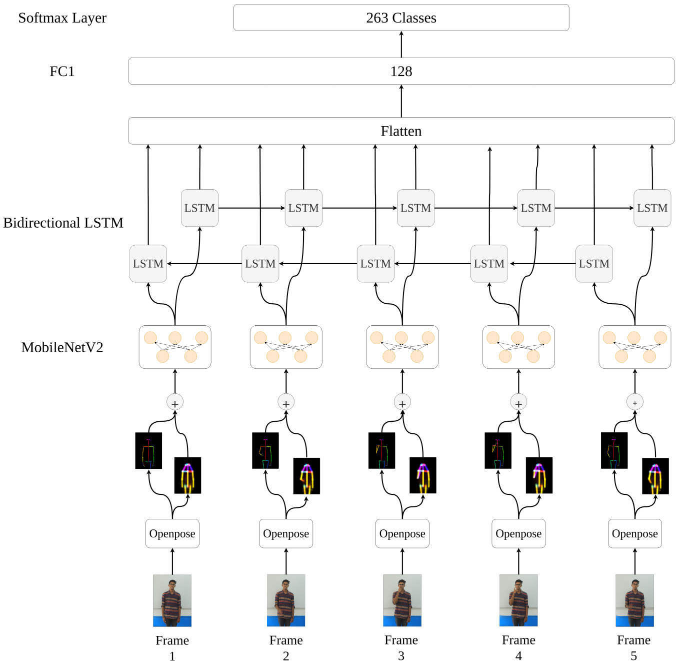
*Method 1 - Using Key-Points Vector.* We extract shoulder, arm, hand and finger key-points for each frame using the pre-trained OpenPose model. In some frames, OpenPose fails to generate key-points. We optionally add an imputation step using the Lucas-Kanade sparse optical flow method [2]. This method fills in the missing key-points assuming a continuity in the spatial movement of each key-point across frames. The feature vector is then flattened across frames and normalised. The feature vector has 96 points per frame, and videos are zero-padded to 200 frames. This results in a feature vector of size 19,200. This vector can then be input to any machine learning model for classification. Various neural networks that perform well on temporal data, such as vanilla RNNs and LSTM networks, yielded poor results on this dataset. Thus, we chose XGBoost [12] as the classifier given its efficiency in inference, good performance across several tasks, and interpretability.

*Method 2 - Using Pose and/or PAF Videos.* We process features extracted as Pose or PAF videos with a two-step network. In the first step, we use a pre-trained deep neural network (DNN) to encode each frame of the video into a latent embedding. In the second step, we process the frame-wise embedding using a sequential model to classify into the different signs.

We consider three options for extracting features: (a) using a Pose frame alone, (b) using a PAF frame alone, and (c) concatenating Pose and PAF frames, thus creating a 2 channel image for each frame as the input. For the encoder, we use four well known CNN backbone networks: MobileNetV2, ResNet50V2, DenseNet201 and InceptionV3. Each of these networks are pre-trained on the ImageNet dataset [16]. In each network, the weights of the pre-trained networks are frozen, the last softmax layer (used for classifying into the ImageNet classes) is deleted, and the output of the penultimate layer is passed through an average-pool layer. The output of this average-pool layer is then the latent embedding for each input

[1] In our experiments we discard the facial and leg key-points and thus have 48 key-points per frame.

**Figure 2: Model pipeline for Method 2 - Each frame is passed through OpenPose and the Pose and PAF videos are obtained. A channel-wise concatenation is done and fed through MobileNetV2 model to extract features. The extracted features are fed to a BiLSTM. The hidden states from LSTM cells are flattened and passed through a fully connected layer and a softmax layer for classification.**



frame. The size of the embedding vector varies depending on the backbone CNN: It is 640 for MobileNetV2, 960 for DenseNet201, and 1024 for ResNet50v2 and Inceptionv3.

The encoder generates a fixed size latent embedding for each frame. The decoder then processes these to generate a class label. After experimenting with different sequence models, we chose a Bidirectional LSTM (BiLSTM) as the decoder. The decoder model has a single Bidirectional LSTM layer with 128 units, the output of which is flattened and fed into a fully-connected layer with 128 units and a softmax layer for classification, with a dropout layer of value 0.4 in between. The model design is captured in Figure 2.

Note that in this method the only trainable parameters are the weights in the decoder. Thus, while the end-to-end network is large, only a small fraction of the weights are fine-tuned for a task-set. If such a method achieves a good accuracy it can enable supporting SLR on multiple sign languages efficiently.

## 5 EXPERIMENTAL EVALUATION

In this section, we report the accuracy of different methods proposed in the previous section on the INCLUDE-50 and INCLUDE datasets. Finally, we also report accuracy of some of our methods for a subset of the ASLLVD dataset.

### 5.1 Results on INCLUDE-50 dataset

Recall that the INCLUDE-50 dataset is a subset of the INCLUDE dataset with 50 signs across 15 categories. We augment the dataset using methods described in the previous section. On average each original video is augmented into 4 additional videos, hence the augmented INCLUDE-50 dataset contains 4,790 videos with approximately 313,290 frames. We process each frame of the augmented dataset by passing it through OpenPose to extract features.

*Training configuration.* For Method 1, XGBoost was trained using 200 gradient boosted trees with a maximum tree depth of 5, base score of 0.5 and learning rate of 0.1. We also note that using optical flow as an imputation method did not result in a better accuracy.

For Method 2 with pose and PAF videos, we train multiple networks, in each case freezing the parameters of the pre-trained backbone CNNs and only training the BiLSTMs. These were trained with the Adam optimizer, a learning rate of 0.0001 for 200 epochs, and a batch size of 64. An L2 regularisation with parameter 0.001 is applied on the weights of the model. Further, an early stopping criterion with patience of 5 epochs was used. A batch normalization [25] layer was added to accelerate training.

*Results.* The results of different methods are reported in Table 3. These results measure the accuracy of a trained model on INCLUDE-50 Test dataset. The best performing model has an accuracy of 94.5% on the 50-class classification task. It uses the Pose video as the extracted feature, a MobileNetV2 backbone, and a BiLSTM decoder. The accuracy compares favourably with related work, where the best reported accuracy for datasets range between 50.6% [34] to 97% [39]. A few more observations can be made from the results:

- Augmentation of the dataset leads to a marked increase in the accuracy. For Method 1 with XGBoost, augmentation increases accuracy by 14% while for Method 2, augmentation increases accuracy by over 20%.
- Using only pose points and the XGBoost classifier provides the second highest accuracy of 89.1%. This enables an efficient inference option using only a short representation of each frame followed by XGBoost which can be executed fast even on standard CPUs. To the best of our knowledge, this is the first demonstration of high accuracy in SLR with the use of XGBoost or related methods. This further strengthens the claim amongst practitioners that XGBoost works well across data-types including temporal data.
- Without augmentation, the XGBoost based classifier has a higher accuracy than with MobileNetV2 based decoder. This comparison reverses with augmentation. This suggests that the DNN based methods are generalising better as the training data-set expands with augmentation.
- Between the usage of Pose and PAF videos for feature extraction, there is no clear choice: Depending on the CNN backbone used for encoding, the results vary.
- Concatenating Pose and PAF videos as features has a higher accuracy, except for the case of MobileNetV2 which gives the highest recorded accuracy with only the Pose video.

*Interpretation of Features.* Though Method 1 was improved upon by the DNN method with augmentation, the XGBoost classifier uses decision trees which are amenable to interpretation. In particular, we ask the question: Which keypoints are important for the identification of the sign, and at what time intervals during signing? This analysis can be performed by computing the relative metric *Gain*, which is the average training loss reduction gained when using a feature for splitting [12]. Higher the Gain value, the higher is the importance of that feature in classification. To calculate the Gain for *hand*, we average the gain value of the finger points and similarly for *arm*, we average shoulder, elbow and wrist points. Thus, with these metrics we tracked the importance of each arm (as the average gain value for the respective shoulder, elbow and wrist key-points) and both hands (as the average gain of all respective fingers). The computed Gain values are shown in Figure 3. The following conclusions may be drawn from these:

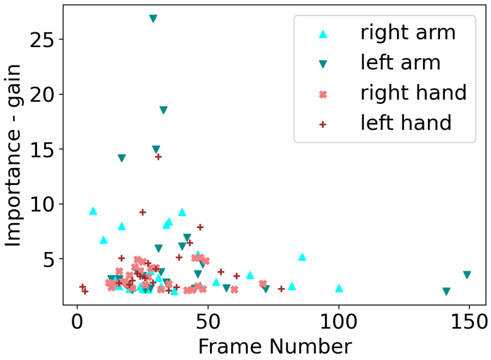
- The features in frames 12 through 75 (0.5 - 3 seconds) have higher importance. This agrees with our intuition as most signs start and end at the same resting point and hence both ends in time are not as important.
- Both hands and both arms are important, though there are small asymmetries. For instance, the left arm has higher gain values around frames 20 to 40, while the right arm has moderate values in either side of this interval. This fits the well known fact that ISL, unlike some other sign languages, is a largely two-handed.
- The information of features beyond 4 seconds is small, and thus we may crop sign videos of isolated signs to 4 seconds from the start of the sign.

**Table 3: Accuracy of different methods on INCLUDE-50**

| Aug | Features | Encoder | Decoder | Acc(%) |
|---|---|---|---|---|
| No | Pose Video + PAF | MobileNetV2 | BiLSTM | 73.9 |
| No | Pose Points | NA | XGBoost | 75.0 |
| Yes | Pose Video | ResNet50V2 | BiLSTM | 74.2 |
| Yes | Pose Video | DenseNet201 | BiLSTM | 80.1 |
| Yes | Pose Video | InceptionV3 | BiLSTM | 85.0 |
| Yes | Pose Points, Imputation | NA | XGBoost | 88.8 |
| Yes | Pose Points | NA | XGBoost | 89.1 |
| **Yes** | **Pose Video** | **MobileNetV2** | **BiLSTM** | **94.5** |
| Yes | PAF | MobileNetV2 | BiLSTM | 78.3 |
| Yes | PAF | ResNet50V2 | BiLSTM | 79.8 |
| Yes | PAF | InceptionV3 | BiLSTM | 83.2 |
| Yes | PAF | DenseNet201 | BiLSTM | 84.4 |
| Yes | Pose Video + PAF | DenseNet201 | BiLSTM | 85.3 |
| Yes | Pose Video + PAF | ResNet50V2 | BiLSTM | 86.8 |
| Yes | Pose Video + PAF | InceptionV3 | BiLSTM | 87.6 |
| Yes | Pose Video + PAF | MobileNetV2 | BiLSTM | 93.3 |

*Effect of dropping frames.* An important limitation of our method is the dependence on pre-trained OpenPose for feature extraction. This significantly reduces training complexity, but at an inference cost; on a high-end NVIDIA V100 GPU, processing a single frame using OpenPose takes around 130 ms. As a first step, we consider an approach to make inference faster by uniformly sampling and selecting frames in the video. We evaluate our best performing encoder model, MobileNetV2 on INCLUDE-50 with this dropping in place. We observe that the accuracy of the model reduces to 92.1% from 94.5% when every alternate frame is selected. Upon changing this to selecting every fourth frame, the accuracy drops to 88.2%, and falls to 82.2% when every eighth frame is selected. Recall that the original video is shot at 25 fps, yet the pipeline has a respectable accuracy of 82.2% at just over 3fps. Thus, dropping frames provides a broad trade-off between inference time and accuracy.

**Figure 3: Relative importance of different key-points from the XGBoost model as measured by the Gain metric.**



## 5.2 Results on the INCLUDE dataset

We now present results of applying our methods to the entire INCLUDE dataset which has 263 classes. These results are shown in Table 4.

The discussed augmentations are applied to the INCLUDE dataset. This increases the total number of videos to over 23,000. Given computation costs, we were unable to train multiple models on this very large dataset. Thus, we report results with augmentation for only the model that had the highest accuracy for INCLUDE-50. We use a larger BiLSTM as our decoder model on this dataset. This achieves our highest recorded accuracy of 85.6%. The following observations can be made:

- Across all models, the accuracy reduces as we move from the 50-class INCLUDE-50 to the 263-class INCLUDE.
- Again, augmentation plays a significant role in increasing the accuracy of the methods. The highest reported accuracy increases by over 15% after augmentation.
- Like in the case of INCLUDE-50, XGBoost based method (only without augmentation) provides comparable performance, though not the highest.

To the best of our knowledge, 85.6% is the highest reported accuracy on any video SLR dataset with 200 or more classes.

## 5.3 Results on American Sign Language

The main focus of this paper is to propose a novel dataset for ISL and benchmark the first SLR models for the dataset. Nevertheless, our best performing model has an interesting feature - most of it is pre-trained. We use the pre-trained OpenPose model for feature extraction and pre-trained CNNs for feature extraction. Only the decoder BiLSTM and the final dense layer are trainable. This motivates

Table 4: Accuracy of different methods on INCLUDE

| Aug | Features | Encoder | Decoder | Acc (%) |
|---|---|---|---|---|
| No | Pose Points | NA | XGBoost | 63.1 |
| No | Pose Video | ResNet50V2 | BiLSTM | 60.5 |
| No | Pose Video | DenseNet201 | BiLSTM | 61.9 |
| No | Pose Video | InceptionV3 | BiLSTM | 69.9 |
| No | Pose Video | MobileNetV2 | BiLSTM | 69.3 |
| No | PAF | MobileNetV2 | BiLSTM | 63.1 |
| No | PAF | ResNet50V2 | BiLSTM | 67.3 |
| No | PAF | InceptionV3 | BiLSTM | 68.0 |
| No | PAF | DenseNet201 | BiLSTM | 70.0 |
| **Yes** | **Pose Video** | **MobileNetV2** | **BiLSTM** | **85.6** |

a design wherein SLR on multiple sign languages can simultaneously be supported: they would share the same feature extractor and encoder, and only specialise in the decoder. To evaluate this, we train our models to perform SLR for the American Sign Language Lexicon Video Dataset (ASLLVD) [47]. ASLLVD has 3300+ classes, but we restrict our analysis to only those classes which have at least 8 videos[2]. There are a total of 48 such classes with 443 total video recordings across them. Each recording contains a front-view video and a side-view video of the subject.

ASLLVD and INCLUDE differ with respect to their frames per second: The ASLLVD dataset was shot at 60 fps, while the processed videos are available at 15 fps. The INCLUDE-50 dataset was shot and is available at 25 fps. The average video in ASLLVD is almost 9s long, while average length for INCLUDE-50 is 2.54s. ASLLVD has a front and a side view video for each recorded sign. INCLUDE contains only front view videos. Notwithstanding these differences, we analysed the applicability of our proposed methods on ASLLVD.

First, we augment the ASLLVD videos with methods similar to the ones discussed in the previous section. We generate 8 variants per video, totalling over 60 front view and 60 side view videos per class on average. Then we pass each frame through OpenPose generating features frame-wise. Note that the front view and side view videos have both been used during training and testing. Method 1 using XGBoost obtains an accuracy of 73.2%. Method 2, with a concatenation of Pose and PAF videos as features, DenseNet201 as the encoder, and a BiLSTM decoder, obtaine an accuracy of 92.1%.

This is the highest reported accuracy on a subset of ASLLVD with 48 or more classes. Given the size of ASLLVD, many researchers have reported results on chosen subsets. A brief summary of the existing results on ASLLVD are as follows: 63.15% on 97 signs [59], 85% on 20 signs [41], 56.82% on 20 signs [15], and 91.58% on 50 signs [22]. The last result [22] uses 3D-CNNs where the entire model is learnt. In contrast, our model has pre-trained networks for feature extraction and encoding. Indeed, the similarity of our results on ASLLVD (92.1% on 48 classes) and INCLUDE-50 (94.5% on 50 classes) provides early evidence that this DL pipeline (with a large part of it fixed) can be used for SLR on multiple languages.

## 6 CONCLUSIONS AND FUTURE WORK

In this paper we present INCLUDE; the only large-scale publicly available dataset on the Indian Sign Language (ISL). The size and quality of the dataset enable exploration of deep models for Sign Language Recognition on ISL. We present a comparison of multiple deep learning models and identify a model that achieves a high accuracy. We also demonstrate that our method achieves state-of-the-art accuracy on an American Sign Language dataset.

We plan to expand this work by collecting more data and building towards automated SLR on a moderate size ISL vocabulary. Further, there are two open challenges that emerge from our work. First, can we optimise the OpenPose network to be more efficient while restricting it to only output features relevant for signing? And second, can we design deep learning models to accurately generalise to signs even with few sample videos per sign?

## ACKNOWLEDGEMENT

We would like to thank the St. Louis College for the Deaf and Blind, Chennai for their constant support and commitment to this project. INCLUDE would not have been possible without them. We also thank Shubham Kanekar for his efforts in recording this dataset.

## APPENDIX

Below is the full list of words in INCLUDE across all categories: **Adjectives**: *loud, mean, rich, poor, thick, thin, expensive, cheap, flat, curved, male, quiet, female, tight, loose, high, low, soft, hard, deep, shallow, clean, happy, dirty, strong, weak, dead, alive, heavy, light, famous, sad, beautiful, Ugly, Deaf, long, short, blind, tall, wide, narrow, big/large, small/little, slow, fast, hot, cold, warm, Nice, cool, new, old, young, good, bad, wet, dry, sick, healthy.* **Animals**: *Dog, Cat, Fish, Bird, Cow, Mouse, Horse, Animal.* **Clothes**: *Hat, Dress, Suit, Skirt, Shirt, T-Shirt, Pant, Shoes, Pocket, Clothing.* **Colours**: *Red, Green, Blue, Yellow, Brown, Pink, Orange, Black, White, Grey, Colour.* **Days and Time**: *Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Today, Tomorrow, Yesterday, Week, Month, Year, Hour, Minute, Second, Morning, Afternoon, Evening, Night, Time, Second (Number).* **Electronics**: *Clock, Lamp, Fan, Cell phone, Computer, Laptop, Screen, Camera, Television, Radio.* **Greetings**: *Hello, How are you, Alright, Good Morning, Good afternoon, Good evening, Good night, Thank you, Pleased.* **Means of Transport**: *Plane, Car, Truck, Bicycle, Bus, Boat, train ticket, Transportation, Train.* **Objects at Home**: *Table, Chair, Bed, Dream, Window, Door, Bedroom, Kitchen, Bathroom, Pencil, Pen, Photograph, Soap, Book, Page, Key, Paint, Letter, Paper, Lock, Telephone, Bag, Box, Gift, Card, Ring, Tool.* **Occupations**: *Teacher, Student, Lawyer, Doctor, Patient, Waiter, Secretary, Priest, Police, Soldier, Artist, Author, Manager, Reporter, Actor, Job.* **People**: *Son, Daughter, Mother, Father, Parent, Baby, Man, Woman, Brother, Sister, Family, Grandfather, Grandmother, Husband, Wife, King, Queen, President, Neighbour, Boy, Girl, Child, Adult, Friend, Player, Crowd.* **Places**: *City, House, Street/Road, Train Station, Restaurant, Court, School, Office, University, Park, Store/Shop, Library, Hospital, Temple, Market, India, Ground, Bank, Location.* **Pronouns**: *I, you, he, she, it, we, you (plural), they.* **Seasons**: *Summer, Spring, Winter, Fall, Season, Monsoon.* **Society**: *Religion, Energy, War, Peace, Attack, Election, Newspaper, Gun, Sport, Exercise, Ball, Death, Price, Sign, Science, God, Medicine, Money, Bill, Marriage, Team, Race (ethnicity), Technology*

[2] Over the years multiple authors have followed different methods to subset the ASLLVD with no clear consensus. ASLLVD also has some special classes where a single sign is repeated multiple times in a video. We ignore these classes in our evaluation.

## REFERENCES

[1] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Quan Yuan, and Ashwin Thangali. 2008. The American Sign Language Lexicon Video Dataset. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 0 (06 2008), 1–8. https://doi.org/10.1109/CVPRW.2008.4563181

[2] Simon Baker and Iain Matthews. 2004. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision* 56, 3 (Feb. 2004), 221–255. https://doi.org/10.1023/B:VISI.0000011205.11775.fd

[3] K. Bantupalli and Y. Xie. 2018. American Sign Language Recognition using Deep Learning and Computer Vision. In *2018 IEEE International Conference on Big Data (Big Data)*. 4896–4899.

[4] Ursula Bellugi and Susan Fischer. 1972. A comparison of sign language and spoken language. *Cognition* 1, 2-3 (Jan. 1972), 173–200. https://doi.org/10.1016/0010-0277(72)90018-2

[5] Yunus Can Bilge, Nazli Ikizler, and Ramazan Cinbis. 2019. Zero-Shot Sign Language Recognition: Can Textual Data Uncover Sign Languages? (07 2019).

[6] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural Sign Language Translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 7784–7793. https://doi.org/10.1109/CVPR.2018.00812

[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv:1812.08008 [cs]* (May 2019). http://arxiv.org/abs/1812.08008 arXiv: 1812.08008.

[8] Cao Dong, Ming C. Leu, and Zhaozheng Yin. 2015. American Sign Language alphabet recognition using Microsoft Kinect. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Boston, MA, USA, 44–52. https://doi.org/10.1109/CVPRW.2015.7301347

[9] Joao Carreira and Andrew Zisserman. 2018. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. *arXiv:1705.07750 [cs]* (Feb. 2018). http://arxiv.org/abs/1705.07750 arXiv: 1705.07750.

[10] C. Charayaphan and A. E. Marble. 1992. Image processing system for interpreting motion in American Sign Language. *Journal of Biomedical Engineering* 14, 5 (1992), 419 – 425. https://doi.org/10.1016/0141-5425(92)90088-3

[11] James Charles, Tomas Pfister, Mark Everingham, and Andrew Zisserman. 2014. Automatic and Efficient Human Pose Estimation for Sign Language Videos. *International Journal of Computer Vision* 110, 1 (Oct. 2014), 70–90. https://doi.org/10.1007/s11263-013-0672-6

[12] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. ACM Press, San Francisco, California, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[13] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. 2017. Sign Language Recognition Using Sub-units. In *Gesture Recognition*, Sergio Escalera, Isabelle Guyon, and Vassilis Athitsos (Eds.). Springer International Publishing, Cham, 89–118. https://doi.org/10.1007/978-3-319-57021-1_3 Series Title: The Springer Series on Challenges in Machine Learning.

[14] Runpeng Cui, Hu Liu, and Changshui Zhang. 2017. Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Cleison Correia de Amorim, David Macêdo, and Cleber Zanchettin. 2019. Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition. *arXiv:1901.11164 [cs, stat]* (Jan. 2019). https://doi.org/10.1007/978-3-030-30493-5_59 arXiv: 1901.11164.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Miami, FL, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[17] Ashish Doval. 2013. The People&#39;s Linguistic Survey of India SIGN LANGUAGE. (2013). https://www.academia.edu/34740076/The_Peoples_Linguistic_Survey_of_India_SIGN_LANGUAGE

[18] Gaolin Fang, Wen Gao, and Debin Zhao. 2007. Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 37, 1 (Jan. 2007), 1–9. https://doi.org/10.1109/TSMCA.2006.886347

[19] S. S. Fels and G. E. Hinton. 1993. Glove-Talk: a neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks* 4, 1 (1993), 2–8.

[20] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. *arXiv:1606.01847 [cs]* (Sept. 2016). http://arxiv.org/abs/1606.01847 arXiv: 1606.01847.

[21] Srujana Gattupalli, Amir Ghaderi, and Vassilis Athitsos. 2016. Evaluation of Deep Learning based Pose Estimation for Sign Language Recognition. In *Proceedings of the 9th ACM International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '16*. ACM Press, Corfu, Island, Greece, 1–7. https://doi.org/10.1145/2910674.2910716

[22] Tao He, Hua Mao, and Zhang Yi. 2017. Moving object recognition using multi-view three-dimensional convolutional neural networks. *Neural Computing and Applications* 28, 12 (Dec. 2017), 3827–3835. https://doi.org/10.1007/s00521-016-2277-9

[23] Eun-Jung Holden, Gareth Lee, and Robyn Owens. 2005. Australian sign language recognition. *Machine Vision and Applications* 16, 5 (Nov. 2005), 312. https://doi.org/10.1007/s00138-005-0003-1

[24] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based Sign Language Recognition without Temporal Segmentation. In *AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA*.

[25] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (March 2015). http://arxiv.org/abs/1502.03167 arXiv: 1502.03167.

[26] Longlong Jing, Elahe Vahdani, Matt Huenerfauth, and Yingli Tian. 2019. Recognizing American Sign Language Manual Signs from RGB-D Videos. *CoRR* abs/1906.02851 (2019). arXiv:1906.02851 http://arxiv.org/abs/1906.02851

[27] Jong-Sung Kim, Won Jang, and Zeungnam Bien. 1996. A dynamic gesture recognition system for the Korean sign language (KSL). *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26, 2 (1996), 354–359.

[28] Hamid Reza Vaezi Joze. [n.d.]. MS-ASL: A Large-Scale Data Set and Benchmark for Understanding American Sign Language. ([n. d.]), 16.

[29] Nayan M. Kakoty and Manalee Dev Sharma. 2018. Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove. *Procedia Computer Science* 133 (2018), 55–62. https://doi.org/10.1016/j.procs.2018.07.008

[30] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-Scale Video Classification with Convolutional Neural Networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Columbus, OH, USA, 1725–1732. https://doi.org/10.1109/CVPR.2014.223

[31] P. V. V. Kishore and P. Rajesh Kumar. 2012. A Video Based Indian Sign Language Recognition System (INSLR) Using Wavelet Transform and Fuzzy Logic. *International Journal of Engineering and Technology* 4, 5 (2012), 537–542. https://doi.org/10.7763/IJET.2012.V4.427

[32] Sang-Ki Ko, Chang Jo Kim, Hyedong Jung, and Choongsang Cho. 2019. Neural Sign Language Translation based on Human Keypoint Estimation. *arXiv:1811.11436 [cs]* (June 2019). http://arxiv.org/abs/1811.11436 arXiv: 1811.11436.

[33] Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding* 141 (Dec. 2015), 108–125. https://doi.org/10.1016/j.cviu.2015.09.013

[34] Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 3793–3802. https://doi.org/10.1109/CVPR.2016.412

[35] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. 2016. Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In *BMVC*.

[36] D. Konstantinidis, K. Dimitropoulos, and P. Daras. 2018. SIGN LANGUAGE RECOGNITION BASED ON HAND AND BODY SKELETAL DATA. In *2018 - 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*. 1–4.

[37] Pradeep Kumar, Himaanshu Gauba, Partha [Pratim Roy, and Debi [Prosad Dogra. 2017. A multimodal framework for sensor based sign language recognition. *Neurocomputing* 259 (2017), 21 – 38. https://doi.org/10.1016/j.neucom.2016.08.132

[38] A. Kuznetsova, L. Leal-Taixé, and B. Rosenhahn. 2013. Real-Time Sign Language Recognition Using a Consumer Depth Camera. In *2013 IEEE International Conference on Computer Vision Workshops*. 83–90.

[39] Simon Lang, Marco Block, and Raúl Rojas. 2012. Sign Language Recognition Using Kinect. In *Artificial Intelligence and Soft Computing*, Leszek Rutkowski, Marcin Korytkowski, Rafa\l Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 394–402.

[40] Kian Lim, Alan Tan, Chin-Poo Lee, and Shing Tan. 2019. Isolated sign language recognition using Convolutional Neural Network hand modelling and Hand Energy Image. *Multimedia Tools and Applications* 78 (02 2019). https://doi.org/10.1007/s11042-019-7263-7

[41] Kian Ming Lim, Alan W. C. Tan, and Shing Chiang Tan. 2016. Block-based histogram of optical flow for isolated sign language recognition. *Journal of Visual Communication and Image Representation* 40 (2016), 538 – 545. https://doi.org/10.1016/j.jvcir.2016.07.020

[42] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Vol. 9905. Springer International Publishing, Cham, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[43] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An End-To-End Deep Video Compression Framework. In *2019*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Long Beach, CA, USA, 10998–11007. https://doi.org/10.1109/CVPR.2019.01126

[44] Zahedi M., Keysers D., Deselaers T., and Ney H. 2005. *Combination of Tangent Distance and an Image Distortion Model for Appearance-Based Sign Language Recognition*. Deutsche Arbeitsgemeinschaft für Mustererkennung Symposium (DAGM), Lecture Notes in Computer Science, Vol. 3663. https://www-i6.informatik.rwth-aachen.de/publications/download/79/Zahedi-DAGM-2005.pdf

[45] Ming-Hsuan Yang, N. Ahuja, and M. Tabb. 2002. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 8 (2002), 1061–1074.

[46] Anup Nandy, Jay Shankar Prasad, Soumik Mondal, Pavan Chakraborty, and G. C. Nandi. 2010. Recognition of Isolated Indian Sign Language Gesture in Real Time. In *Information Processing and Management (Communications in Computer and Information Science)*, Vinu V. Das, R. Vijayakumar, Narayan C. Debnath, Janahanlal Stephen, Natarajan Meghanathan, Suresh Sankaranarayanan, P. M. Thankachan, Ford Lumban Gaol, and Nessy Thankachan (Eds.). Springer, Berlin, Heidelberg, 102–107. https://doi.org/10.1007/978-3-642-12214-9_18

[47] Carol Neidle, Ashwin Thangali, and Stan Sclaroff. 2012. Challenges in development of the American Sign Language Lexicon Video Dataset (ASLLVD) corpus. https://open.bu.edu/handle/2144/31899

[48] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. 2015. Sign Language Recognition Using Convolutional Neural Networks. In *Computer Vision - ECCV 2014 Workshops*, Lourdes Agapito, Michael M. Bronstein, and Carsten Rother (Eds.). Springer International Publishing, Cham, 572–578.

[49] Lionel Pigou, Mieke Van Herreweghe, and Joni Dambre. 2017. Gesture and Sign Language Recognition with Temporal Residual Networks. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, Venice, 3086–3093. https://doi.org/10.1109/ICCVW.2017.365

[50] Elakkiya R. and K. Selvamani. 2017. Extricating Manual and Non-Manual Features for Subunit Level Medical Sign Modelling in Automatic Sign Language Classification and Recognition. *Journal of Medical Systems* 41 (11 2017). https://doi.org/10.1007/s10916-017-0819-z

[51] J. L. Raheja, Anand Mishra, and Ankit Chaudhary. 2016. Indian Sign Language Recognition Using SVM. *Pattern Recognition and Image Analysis* (June 2016). https://doi.org/10.1134/S1054661816020164

[52] J. Rekha, J. Bhattacharya, and S. Majumder. 2011. Shape, texture and local movement hand gesture features for Indian Sign Language recognition. In *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*. IEEE, Chennai, India, 30–35. https://doi.org/10.1109/TISC.2011.6169079

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]* (Jan. 2016). http://arxiv.org/abs/1506.01497 arXiv: 1506.01497.

[54] Yogeshwar Rokade and Prashant Jadav. 2017. Indian Sign Language Recognition System. https://www.researchgate.net/publication/318656956_Indian_Sign_Language_Recognition_System

[55] Rung-Huei Liang and Ming Ouhyoung. 1998. A real-time continuous gesture recognition system for sign language. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. 558–567.

[56] T. Starner, J. Weaver, and A. Pentland. 1998. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 12 (Dec. 1998), 1371–1375. https://doi.org/10.1109/34.735811

[57] Chao Sun, Tianzhu Zhang, Bing-Kun Bao, and Changsheng Xu. 2013. Latent support vector machine for sign language recognition with Kinect. In *2013 IEEE International Conference on Image Processing*. 4190–4194. https://doi.org/10.1109/ICIP.2013.6738863 ISSN: 2381-8549.

[58] Alistair Sutherland, George Awad, and Junwei Han. 2013. Boosted subunits: a framework for recognising sign language from videos. *IET Image Processing* 7, 1 (Feb. 2013), 70–80. https://doi.org/10.1049/iet-ipr.2012.0273

[59] Stavros Theodorakis, Vassilis Pitsikalis, and Petros Maragos. 2014. Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition. *Image and Vision Computing* 32, 8 (Aug. 2014), 533–549. https://doi.org/10.1016/j.imavis.2014.04.012

[60] Kumud Tripathi and Neha Baranwal G.C. Nandi. 2015. Continuous Indian Sign Language Gesture Recognition and Sentence Formation. *Procedia Computer Science* 54 (2015), 523–531. https://doi.org/10.1016/j.procs.2015.06.060

[61] D. Uebersax, J. Gall, M. Van den Bergh, and L. Van Gool. 2011. Real-time sign language letter and word recognition from depth data. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 383–390.

[62] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H. S. Torr. 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. *arXiv:1812.05050 [cs]* (May 2019). http://arxiv.org/abs/1812.05050 arXiv: 1812.05050.

[63] Zenith Media. 2019. Online video viewing to reach 100 minutes a day in 2021. https://www.zenithmedia.com/online-video-viewing-to-reach-100-minutes-a-day-in-2021/