Intelligent Gesture Recognition System for Translating Indian Sign Language to English

Anagha Kulkarni[1*], Yael Robert[1†], Yashshree Nigudkar[1†], Pranjali Barve[1†] and Namita Mutha[1†]

[1]Department of Information Technology, Cummins College of Engineering for Women, Karvenagar, Pune, 411052, Maharashtra, India.*

**Corresponding author(s). E-mail(s): yael.a@cumminscollege.in;*

Contributing authors: anagha.kulkarni@cumminscollege.in; yashshree.nigudkar@cumminscollege.in; pranjali.barve@cumminscollege.in; namita.mutha@cumminscollege.in;

[†]These authors contributed equally to this work.

Abstract

Sign languages involve a combination of hand movements and facial gestures. Alphabets and digits form static signs whereas dynamic signs consist of words and sentences. Based upon the cultural differences and regional variations, different signs have evolved for a word in each sign language. In reality, every sign language has its own set of signs for each word. As a result, recognizing words and phrases in sign languages is difficult. The recognition of spatial and time-distributed features of Indian Sign Language is the focus of this research. The main goal of this work is to identify gestures in Indian Sign Language using a multi-class classification technique. Various experiments have been conducted using Convolutional Neural Network, Long-Short Term Memory, and Gated Recurrent Units. Processing videos posed challenges. Various experiments and the methodology yielded an accuracy of 87.5% on unseen test data. The most significant advantage of this system is that it does not require any special device such as a depth-sensing camera, hand gloves, or special t-shirts.

Keywords: **Indian Sign Language Recognition, Video Processing, Convolutional Neural Network, Gated Recurrent Units, Deep Learning**

1. Introduction

Evolution of communication is a continuous process. Communication is necessary for the survival of human beings. Communication helps us understand each other. Verbal communication is an important ability of human beings which helps us share information, views, and ideas orally. However, not every human being can talk.

Across the globe, a few million people face challenges while talking with others due to a lack of speaking and listening abilities. According to the data given by the World Federation of the deaf, 70 million people in the world use Sign Language (SL) for communication [1]. However, verbally impaired people, also called signers, across the globe do not use universal SL. Many SLs have evolved across the globe. A study states that there are more than 300 different types of SLs used throughout the world today [1]. In India, Indian Sign Language (ISL) is commonly used by the verbally impaired. Over a period of time, ISL has evolved. Currently, there are 10,000 words in ISL across six categories. ISL contains words required for daily use, academic, legal and administrative words, medical, technical and agricultural words.

SL consists of manual and non-manual gestures or signs (nodding, shrugging, face expressions, etc). Some signs, called static signs, do not need a movement of hands. However, dynamic signs need to be expressed with the movements of hands and fingers. Static SL is generally used for signing alphabets, digits, and a few words. On the other hand, dynamic SL is used to express most of the words, phrases, and sentences.

Different signs are used to represent the same letters, digits, and words in SLs used across the globe. Every SL has its grammar which is different from spoken languages [2, 3]. A SL may have different dialects depending upon the culture of the region. So, recognizing SL is challenging for those who are not familiar with the language. Figure 1 represents the signs used for representing the alphabet 'A' in American SL (ASL), British SL (BSL), French SL (FSL), Chinese SL (CSL), and ISL.

Figure 1. Alphabet 'A' in ASL, BSL, FSL, CSL, ISL

Some SLs use both hands to represent a few alphabets. It can be seen from figure 1 that BSL and ISL use two hands to represent 'A'. Naturally, the signs used for words are non-identical in different SLs. For example, the word 'mother' is signed with an open hand, held near the mouth in ASL whereas in ISL, 'mother' is gestured using an index finger pointing at the nose to show 'nose ring' which is commonly used in the southern part of India [3]. Israeli SL (IsSL) uses the movement of the index finger from one cheek to another [4]. Figure 2 shows these signs.

Figure 2. Word 'mother' in ASL, IsSL, and ISL

SL is a concept-based language. The sign can change based on the region, age of the signer, race, or gender. Hence it is pretty common to have a word that has different signs. For instance, 'mother' in ISL can be expressed in two ways as shown in figure 3. This is because some dialects in the northern part express 'mother' by pointing at 'bindi' on the forehead.

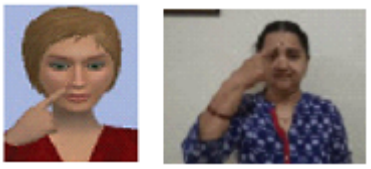


Figure 3. Word 'mother' in ISL

ISL also contains some words which have almost similar signs. For example, the alphabets 'm' and 'n' are pretty identical as shown in figure 4. It is seen that the only difference between the two alphabets is that the index finger, middle finger, and ring finger are used in 'm' whereas in 'n' only the index finger and middle finger are placed on the other hand [5]. Similarly, many words have similar signs. For example, the signs for 'mail' and 'promise' are similar [6]. Figure 5 shows these signs. The difference is that for signing 'mail' the hands are slightly crossed whereas for 'promise' the hands are put on top of each other.

Figure 4. Alphabets 'm' and 'n' respectively in ISL

Figure 5. Signs for 'mail' and 'promise'

Nowadays, translation systems for spoken languages are on the rise [7–9]. This raises the inevitable concern for similar systems to address the problem of the unspoken (non-verbal) language - 'sign language'. Although some research has been done in this domain, full-fledged systems for the signing community are far from achieved. A few systems have been developed which translate SL to text or audio. This paper summarizes the results of experiments on the translation of ISL gestures to English words. The next section presents a literature review. Section 3 presents the methodology. Experiments and results are discussed in section 4. The conclusion is presented in section 5.

2. Literature survey

D. Li et. al focused on increasing the ASL dataset of words [10]. Deep learning techniques were implemented to evaluate the performance. A new pose-based temporal graph convolutional network (Pose-TCGN) was proposed. PoseTCGN uses You Only Look Once (YOLO) based bounding boxes to reduce the effect of the background. The researchers used trajectories to represent the temporal motions. Variations in the dataset were created by adding dialects.

R. Rastoo et. al. proposed a model using Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) [11]. The researchers propose a hand-pose aware model. Pixel level, flow, deep hand, and features from posed hands are inputted to LSTM.

J. Huang et al. proposed a SLR system that extracts spatial and temporal features using 3D CNN [12]. The researchers built their dataset of 25 signs using a depth-sensing Kinect camera. The Kinect camera helped capture the body movements along with the color and depth variations. Every video of signs captured by the Kinect camera was broken into nine frames of size 64 x 48. These frames were processed by CNN and classified further. This work showed that to capture the hand and finger movements, the third dimension in the form of depth holds a good amount of information and helps segregate signs which are similar but mean different words. A depth-sensing camera can easily capture the difference between signs 'mail' and 'promise' as seen in figure 5.

To recognize hand gestures S. Jiang and Y. Chen proposed a two-way hand gesture recognition model in which 3D CNN will learn Spatio-temporal features for LSTM [13]. They also proposed the use of the Adam optimizer to speed up the training process as opposed to Stochastic Gradient Descent (SGD). The use of Adam optimizer with CNN and LSTM proved useful in recognizing signs.

LSTM has proved to be beneficial in capturing temporal features in the gestures [14]. The position and the trajectory followed by the four joints (left hand, right hand, left elbow, and right elbow) helped the researchers in collecting contextual information. Information regarding other joints (such as head, neck, etc.) was of less use according to the researchers. 12-dimensional feature vectors were extracted from the trajectories of four joints, as 3D coordinates in space, and processed by LSTM along with two fully connected layers. The 12-dimensional feature vectors are the input. Joints and their trajectories have helped the researchers in understanding the relative positions of the fingers and hands. The results have been beneficial for CSL.

Researchers segregate gestures from videos based on their hand location [15]. The researchers have tested their model on the Continuous Gesture Dataset. The

model recognizes the gestures from continuously running video. The model uses two streams: one stream uses RGB image to extract color feature maps and the other uses depth image to extract depth feature maps. The outputs of both streams are concatenated for classification.

Xi Hu et al. proposed a method that produced skeletons of hands [16]. Based upon the pose in the skeleton, the gesture in CSL was recognized. Two CNNs were used for generating the hand skeletons. Using Residual Network, the videos were scanned in both forward and backward directions. As the gestures are performed rapidly, researchers realized that some gestures are overlapping which was difficult for the model to handle. After removing the overlaps, the accuracy improved.

3D CNNs were used to extract spatial and temporal features of the video [17]. The Kinect camera was used to capture videos. Contours of images were given more importance while processing the videos. The dataset included 20 Chinese most commonly used words with a total of 6800 videos. 3D kernel temporal was used for 3D convolutions. The feature maps obtained after convolutions were useful to capture spatial information.

A literature survey is presented on static and dynamic ISL [18–21]. The papers summarize various techniques employed to capture images and videos of different SLs across the world. The papers also have presented a survey of different machine and deep learning techniques used to recognize SL alphabets, digits, and words.

ISL recognition research is reported in [22]. The researchers have used the Inception model to recognize the ISL gestures. The model processes input images. The model gave an accuracy of 93%.

Researchers from all over the world are experimenting on techniques and algorithms for making Sign Language Recognition (SLR) as robust as possible. For instance, the authors of [23] worked on a novel strategy for handling subject variability in sensor-based SLR systems. Wireless sensors that record surface Electromyogram (sEMG) and Inertial Measurement Units (IMU) signals are used to collect data for the recognition of Indian Sign Language, but the sEMG signals vary greatly due to sensor placement and arm muscle physiology. To address this issue, the authors proposed Personalized Sign Language Recognition (PSLR), which retrains the deep learning model using a little amount of input from a new user. The accuracy of this transfer learning principle yielded 95.6%.

The system proposed in [24] explores the recognition of ISL static sign images which can also tackle identifying two-hand gestures. This is made possible by the CNN using Depthwise Separable Convolution (DSC) model, which primarily uses CNN for key feature extraction and DSC to cut down on computational costs. The efficiency of the system is tested on custom ISL dataset and publicly available ASL dataset. The authors of the research used data augmentation techniques like rotation, shearing, and skewing in order to generalize the model and increase the intra-class variance.

In their assessment of SLR systems for several languages, Nimratveer and Rajneesh [25] evaluated the advantages and disadvantages of various data collecting techniques such as glove-based, kinect-based leap motion control, and vision. Although a vision-based approach is less expensive and can also cover facial expressions, there are still numerous obstacles to be addressed. The authors also emphasized the requirement for a benchmarked ISL dataset, which is a barrier

for researchers trying to create a comprehensive and precise recognition system. In addition, this review develops a taxonomy that is cohesive and separated into three levels to represent contemporary research: Level 1 Elementary level (Recognition of sign characters), Level 2 Advanced level (Recognition of sign words), and Level 3 Professional level (Sentence interpretation). Hand tracking of both hands for feature extraction, non uniform or non plain backgrounds, signer speed variation, simultaneous recognition of facial expressions, adjusting vocabulary to form meaningful sentences, absence of signs to indicate beginning or end of a sentence are some of the challenges that the authors have shed light on.

Recognizing a word or words from videos is a difficult process as compared to identifying an alphabet or digit from a still image. This is because the movements of a hand or hands and facial expressions are crucial in communicating the word. Furthermore, the hand movements may be fast, making boundary identification problematic. The proposed system attempts to solve these challenges.

Researchers in [26] worked on signal processing methods for automatic sign language recognition as opposed to videogrammetry or Artificial Neural Networks. They have proposed to use the muscle fiber activation and body acceleration in the three axes otherwise known as Surface Electromyography (sEMG) and Accelerometry (ACC) respectively. The corresponding devices were connected to the signer's specific hand muscles and there is a single pose to start with and come back to at the end of the sign. Their target was to classify twelve Columbian signs, custom made, for which accuracy obtained was 96.66%. The sEMG and ACC signals were segmented using Pareto Optimality after which signal features like Permutation Entropy and Root Mean Square (RMS) were assessed. Finally, for classification, Support Vector Machines (SVMs) were employed with the radial basis function kernel and hyperparameter tuning was done using grid search. Since the one-vs-one decision function was finalized, on increase in vocabulary the complexity will increase as the number of classifiers will rise exponentially. Moreover, since there may be initial hand movements unaccounted for, the recognition system gets a segmentation error of 5.8%.

Going back to video based recognition, in [27], deep learning techniques have been applied to solve continuous sign language complexities. These researchers have applied multi-modal data fusion strategies by combining RGB and skeletal inputs. The input video data is clipped using a sliding window technique. Vision Transformer Network is used to get RGB clips' spatio-temporal features while feature extraction for skeletal clips is done using Attention-enhanced Multi-scale 3D Graph Convolutional Network (AM3D-GCN). After that a sign language encoder network based on Transformers, which can learn long-term dependencies, is introduced. Finally, to provide a wholesome meaning of the continuous sign, a Connectionist Temporal Classification (CTC) decoder network is used. This network has been tested on the famous datasets SLR-100 [28] and PHOENIX-Weather 2014T (RWTH) [29]. The word error rates for the SLR-100 dataset and the RWTH PHOENIX-Weather dataset are 1.9%, and 22.8% respectively.

Researchers Akansha Tyagi et al [30] have worked towards effective Indian Sign Language recognition. Unlike the previous works, this one has focused on static signs. The custom dataset includes 24 alphabets and 10 digits. These are a collection of single-handed sign gestures. The proposed system is a feature extraction technique formed by combining features from Fast Accelerated Segment

Test (FAST) and Scale-Invariant Feature Transformation (SIFT) to later be classified by Convolution Neural Networks (CNN). Dataset accuracies of 97.89% for ISL-alphabets, 95.68% for Modified National Institute of Standards and Technology (MNIST) database [31], 94.90% for Jochen Trisech Dataset (JTD) [32], and 95.87% for National University of Singapore dataset II (NUS-II) [33].

3. Methodology

The proposed system is composed of five stages:

1. Training and Validation Dataset Creation
2. Augmentation of Dataset
3. Preprocessing the Videos
4. Test Dataset Creation
5. Classification

3.1 Training and Validation Dataset Creation

As there was no suitable ISL dataset, a dataset was created. The dataset includes videos of the 6 most frequently used ISL terms (which are used as classes for classification). Videos for 'sad', 'happy', 'perfect', 'beautiful', 'dry', and 'deaf' were collected from friends and family using cameras in mobile phones and laptops. 789 videos were collected for these ISL words. In a variety of lighting, attire, and settings, 75 people signed the words to form a part of the training dataset. The videos were 2-4 seconds long depending on the gesture. The details are summarized in Table 1.

Table 1. Details of training and validation datasets

Details	Number
Number of ISL words	6
Number of ISL words	75
Number of videos	789
Video length	2-4 seconds

3.2 Augmentation of dataset

As the collected dataset had only 789 videos, augmentation techniques were applied to create a large size dataset. The augmentation techniques employed included the addition of Gaussian noise, sharpening, applying linear contrast, rotating images, and combining some of these techniques. As a result of augmentation, the new dataset evolved and its size increased by 1578 videos. Thus, the dataset contained 2367 videos. These 2367 videos were split into training and validation sets. Table 2 summarizes the details.

Table 2. Details of augmented dataset

Details	Number
Number of videos after augmentation	2367

3.3 Preprocessing the Videos

The training and validation datasets videos were segmented into frames (or images). It was observed that if more frames per video were segmented, the classification was accurate and precise. However, the time required for processing more frames was more. Hence, 10-15 frames were segmented from every video regardless of its length. Figure 6 shows some frames of the video 'sad'.

Figure 6. Frames segmented from video of 'sad'

Every frame was resized to 224x224 pixels and normalized for further processing. For implicit feature extraction, CNN was used to detect the motion of hands. A custom data generator was used for efficient computing. Table 3 shows the details.

Table 3. Details of preprocessing

Details	Number
Number of frames per video	10-15
Frame size	224x224

3.4 Test dataset creation

To create the test dataset, 4 new people recorded 40 videos signing the above mentioned 6 ISL words. Table 4 shows these details.

Table 4. Details of training and validation datasets

Details	Number
Number of signers	4
Number of videos	40

3.5 Classification

Experiments were conducted with different models and hyperparameters. The frames segmented from augmented training and validation datasets (2367 videos) were inputted to the variants of CNN. Deep neural networks were built which outputted the class name (for example, 'sad', 'happy', and so on) for an entirely new set of 40 videos created for testing purposes. After several experiments, the final training and testing accuracies were 96% and 87.5% respectively. Figure 7 shows the block diagram.

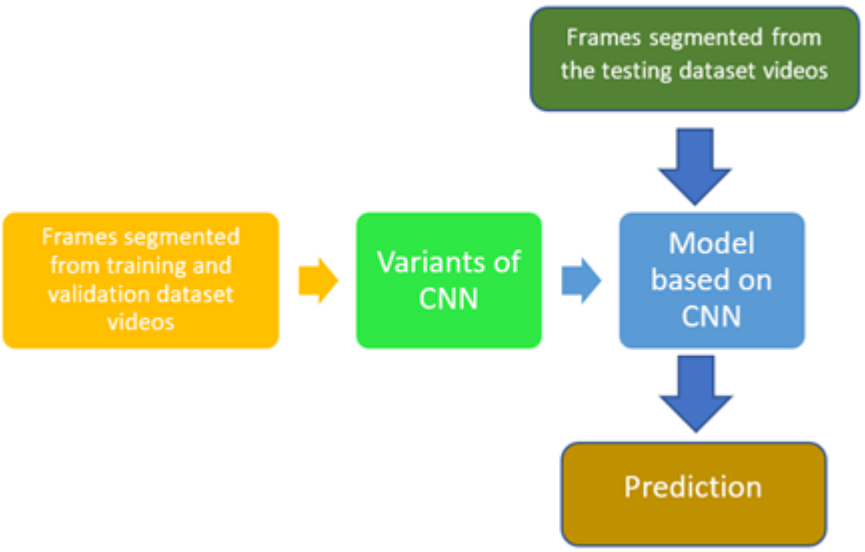


Figure 7. Block diagram of the architecture

4. Experiments and results

After careful research and experimentation, the best model was built for using MobileNet CNN + GRU. The experiments that were conducted are elaborated in this section.

4.1 Building the models

The final model for the dataset containing 2367 videos has evolved as a result of a series of experiments. The results are presented below.

4.1.1 Model 1 (Vanilla CNN):

The first model was built using 10 frames extracted from ISL videos. Figure 8 shows the details. The model was built using 5 Conv2D and max-pooling layers. All the outputs were collated using flatten and dense layers. It can be seen from the model that there are 6 million total parameters and all are trainable. The training accuracy was 18%. The model was overfitting and was not able to handle the transitions and hand movements in the videos. The testing accuracy obtained was 16.6%. Although, in general, CNN works well with images, the model did not work well with frames of ISL videos. The reason was that the

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 26, 26, 128)	0
conv2d_3 (Conv2D)	(None, 24, 24, 256)	295168
max_pooling2d_3 (MaxPooling2	(None, 12, 12, 256)	0
conv2d_4 (Conv2D)	(None, 10, 10, 512)	1180160
max_pooling2d_4 (MaxPooling2	(None, 5, 5, 512)	0
conv2d_5 (Conv2D)	(None, 3, 3, 1024)	4719616
max_pooling2d_5 (MaxPooling2	(None, 1, 1, 1024)	0
flatten (Flatten)	(None, 1024)	0
dropout (Dropout)	(None, 1024)	0
dense (Dense)	(None, 6)	6150

Total params: 6,294,342
Trainable params: 6,294,342
Non-trainable params: 0

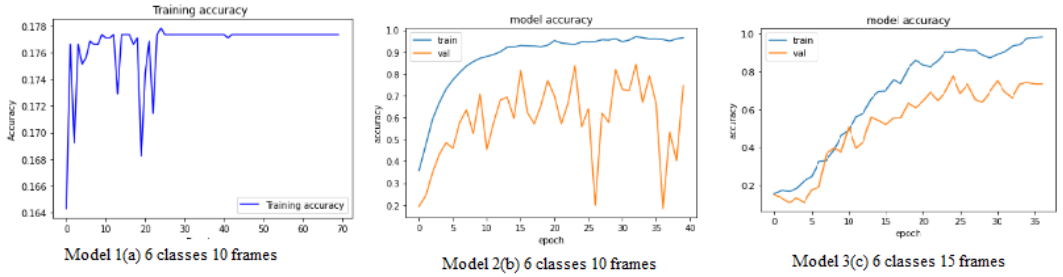
Figure 8. Vanilla CNN model

subsequent frames were time-dependent on each other. CNN failed to capture the relationships between the frames. For example, the video for 'sad' was divided into 10 frames. Each frame was labeled 'sad'. When 10 frames were inputted to the vanilla CNN, it couldn't capture the dependencies between the frames and hence could not classify the videos accurately. In reality, the 10 frames were to be processed as a series. The training accuracy graph for this model can be seen in figure 9 (a).

4.1.2 Model 2 (CNN + LSTM):

The second model was built using CNN+LSTM. CNN was used for feature extraction and LSTM was used to handle the time-distributed nature of the data. A custom data generator was built to process 10 frames per video. For processing videos in model 1, all the 10 frames (belonging to a video) were labeled

individually. As opposed to this, for model 2, 10 frames (belonging to a video) were labeled together as a group. This ensured that the model processed the groups of frames in a time-distributed manner. The training



Model 1(a) 6 classes 10 frames Model 2(b) 6 classes 10 frames Model 3(c) 6 classes 15 frames

Figure 9. Different architecture-based model variations and corresponding graphs

Layer (type)	Output Shape	Param #
time_distributed (TimeDistri	(None, 335, 5, 5, 128)	589952
time_distributed_1 (TimeDist	(None, 335, 5, 5, 64)	73792
time_distributed_2 (TimeDist	(None, 335, 2, 2, 64)	0
time_distributed_3 (TimeDist	(None, 335, 2, 2, 64)	36928
time_distributed_4 (TimeDist	(None, 335, 2, 2, 32)	18464
time_distributed_5 (TimeDist	(None, 335, 1, 1, 32)	0
time_distributed_6 (TimeDist	(None, 335, 1, 1, 32)	128
time_distributed_7 (TimeDist	(None, 335, 32)	0
dropout (Dropout)	(None, 335, 32)	0
lstm (LSTM)	(None, 32)	8320
dense (Dense)	(None, 64)	2112
dense_1 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 8)	264

Total params: 732,040
Trainable params: 731,976
Non-trainable params: 64

Figure 10. CNN + LSTM model

accuracy increased to 80% whereas validation accuracy was 50%. However, as it can be observed from figure 9 (b), the graph of validation accuracy is very spiky and uneven. The testing accuracy was 17%. After analysis, it was inferred that CNN could extract the features well but the LSTM caused the validation accuracy to be spiky. As the training and validation accuracies improved, the customized data

generator is used in the subsequent model. Figure 10 shows model 2. It can be seen that the model has 64 non-trainable parameters.

4.1.3 Model 3 (MobileNet CNN + GRU):

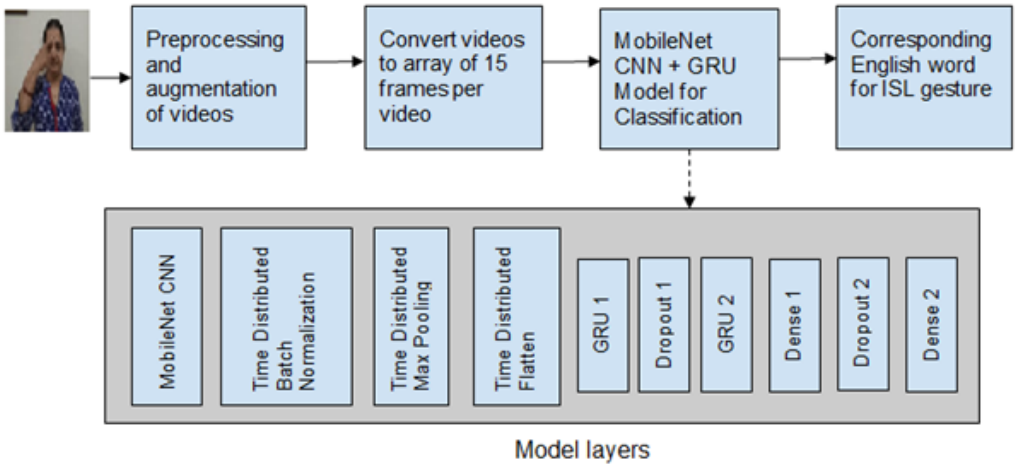


Figure 11. Pipeline of Gesture Recognition for ISL videos

Layer (type)	Output Shape	Param #
time_distributed (TimeDistri	(None, 15, 3, 3, 1024)	3228864
time_distributed_1 (TimeDist	(None, 15, 3, 3, 1024)	4096
time_distributed_2 (TimeDist	(None, 15, 1, 1, 1024)	0
time_distributed_3 (TimeDist	(None, 15, 1024)	0
gru (GRU)	(None, 15, 128)	443136
dropout (Dropout)	(None, 15, 128)	0
gru_1 (GRU)	(None, 64)	37248
dense (Dense)	(None, 150)	9750
dropout_1 (Dropout)	(None, 150)	0
dense_1 (Dense)	(None, 6)	906

Total params: 3,724,000
Trainable params: 3,700,064
Non-trainable params: 23,936

Figure 12. MobileNet CNN + GRU model

Figure 12: In model 3, instead of using vanilla CNN, MobileNet CNN was used. MobileNet CNN is pre-trained on the ImageNet dataset [34, 35]. Every video was divided into 15 frames. A custom data generator was created to feed a batch as an input. 18 unique videos were randomly chosen, as a batch, from the training dataset. As a result, the model had 270 frames in a batch (18 videos*15 frames/video). The advantage of using a pre-trained model was that there was no need to start training from scratch. As the video data was compute-intensive, MobileNet CNN helped in processing the data faster. MobileNet CNN is a simple and lightweight version of CNN. This was followed by batch normalization, max-pooling (2*2 filters), and flattened layers. This process was done in a time-distributed manner. Finally, all the outputs were collated (using flatten). In addition to this, GRU, a variant of Recurrent Neural Network (RNN) and LSTM, was used. Two GRU layers are used, having 128 and 64 units respectively. GRU layers were separated by a dropout layer having a 40% dropout rate. GRU helped in solving the vanishing gradient and overfitting problems. Finally, two dense layers were used with one more dropout layer sandwiched between the dense layers. Softmax activation function was used at dense layers. Various experiments were conducted with different optimizers like Adam, AdaGrad, and AdaDelta of which Adam optimizer gave the best results [36–38]. The pipeline of gesture recognition for ISL videos is shown in figure 11.

The training accuracy increased to 96% and validation accuracy to 80%. Both the graphs were along the same trajectory. This also suggested that the model was a perfect fit. The testing accuracy was 87.5%. The graph of training and validation accuracy is shown in figure 9 (c). Overall, it can be concluded that the training and validation accuracies have improved by a large amount in model 3. After careful analysis, we concluded that vanilla CNN was not able to identify spatial features. LSTMs were able to identify some spatial features, while MobileNet CNN + GRU worked best in identifying spatial features from the video. Model 3 is shown in figure 12.

Figure 13 shows the graphs plotted for different experiments carried out using 10 and 15 frames with and without data augmentation. It can be seen that the training and validation accuracy graphs are closer to each other for 10 and 15 frames with augmentation than for those without augmentation. Thus, we can conclude that augmentation techniques have helped in generalizing the frames. As a result, overfitting has been reduced.

4.2 Fitting the model

Training the model with a large number of videos at a time can cause the system to crash. Hence, the final model was built with entire validation data and 1/6th of training data at a time for 100 epochs. These parameters were adjusted according to the learning rate of every epoch in experimentation.

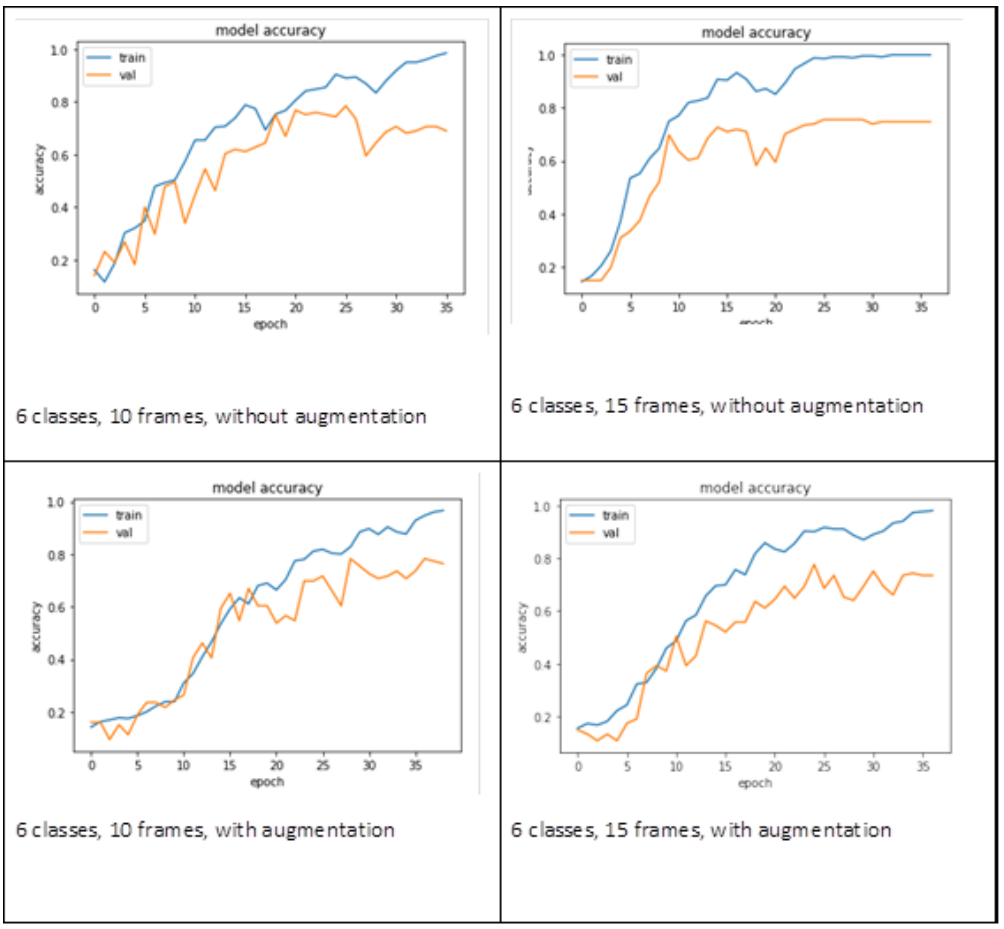


Figure 13. Graphs for CNN + GRU based architecture

4.3 Controlling overfitting

Experiments were conducted to control/avoid overfitting in two ways:

1. EarlyStopping: It helped in monitoring validation loss [39]. The model stops learning when the validation loss stagnates or starts increasing suddenly (which is a sign of overfitting). EarlyStopping helped reduce overfitting to some extent
2. ReduceLROnPlateau: It helped in controlling overfitting [39]. If the model stops improving validation loss within two epochs, the learning rate is reduced by ReduceLROnPlateau. As the learning rate reduces, the model stops learning from noise. Thus, validation loss is improved, thereby, reducing overfitting. ReduceLROnPlateau gives better flexibility and control over other parameters. Our experiments reduced the learning rate by 0.1, starting with a minimum learning rate (1e-8) and patience (6). This gave a reasonable control of how the learning rate should be reduced so that the model can train better. It was observed that a very high or very low learning rate harms the model instead of training. Figure 14 shows the use of ReduceLROnPlateau.

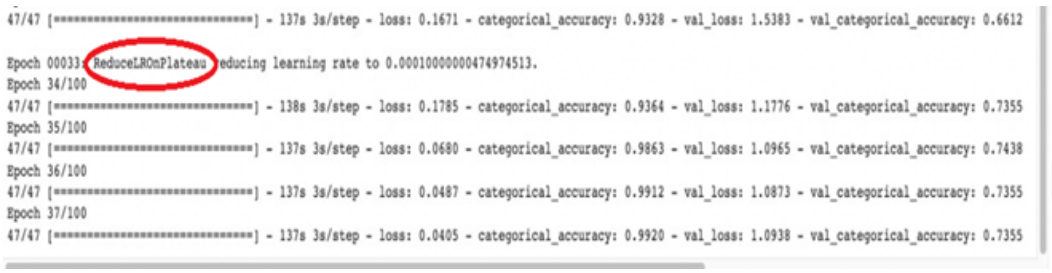


Figure 14. Reduction in Learning Rate using ReduceLROnPlateau

4.4 Comparison

Table 5 shows the comparison of all three models.

Table 5. Comparison of Model 1, Model 2, and Model 3

Model	Number of frames	Training Accuracy	Validation accuracy	Testing accuracy
Model1: Vanilla CNN	10	18%	-	16.6%
Model2: CNN+LSTM	10	80%	50%	17%
Model3: MobileNet CNN+GRU	15	96%	80%	87.5%

5. Conclusion

As the ISL dataset was not available, the dataset was built. ISL videos have spatial and time-distributed features. From the experiments that were conducted, it can be concluded that vanilla CNN is not very effective in capturing spatial and time-distributed features. CNN + LSTM work better than vanilla CNN and can capture the time-distributed nature of the dataset. However, the model was not stable. The customized data generator helped in improving training and validation accuracies. MobileNet CNN + GRU were the most suitable models for the ISL dataset as they captured the spatial and timedistributed nature of the dataset well. The training and validation accuracies were 96% and 80% respectively. The testing accuracy was 87.5%. Data augmentation helped in making the model less sensitive towards lighting conditions, hand angles, and backgrounds, thus helping in generalizing, thereby, reducing overfitting. The use of a customized data generator has helped in the efficient utilization of GPU and RAM. It also helped in reducing the overall prediction time. ReduceLROnPlateau and EarlyStopping gave the model more flexibility and control over parameters. They also helped in avoiding overfitting.

REFERENCES

[1] Nations, U.: International day of sign languages 23 September. United Nations https://www.un.org/en/observances/sign-languages-day (2019).

[2] Stokoe, W.C.: Sign language versus spoken language. Sign Language Studies 18(1), 69–90 (1978).

[3] Hands, T.: Indian Sign Language Resource. http://talkinghands.co.in/ (2021).

[4] Sandler, W.: The body as evidence for the nature of language. Frontiers in psychology 9, 1782 (2018).

[5] ISLRTC: Indian Sign Language Research and Training Centre. http:// islrtc.nic.in/ (2021).

[6] Ansari, Z.A., Harit, G.: Nearest Neighbour classification of indian sign language gestures using Kinect camera. Sadhana 41(2), 161–182 (2016).

[7] Natu, I., Iyer, S., Kulkarni, A., Patil, K., Patil, P.: Text translation from hindi to english. In: International Conference on Advances in Computing and Data Sciences, pp. 481–488 (2018). Springer.

[8] Sharma, V.K., Mittal, N., Vidyarthi, A.: Context-based translation for the out of vocabulary words applied to hindi-english cross-lingual information retrieval. IETE Technical Review, 1–10 (2020).

[9] Laskar, S.R., Khilji, A.F.U.R., Pakray, P., Bandyopadhyay, S.: Hindimarathi cross lingual model. In: Proceedings of the Fifth Conference on Machine Translation, pp. 396–401 (2020).

[10] Li, D., Rodriguez, C., Yu, X., Li, H.: 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, pp. 1459–1469 (2020).

[11] Rastgoo, R., Kiani, K., Escalera, S.: Hand pose aware multimodal isolated sign language recognition. Multimedia Tools and Applications 80(1), 127– 163 (2021).

[12] Huang, J., Zhou, W., Li, H., Li, W.: Sign language recognition using 3d convolutional neural networks. In: 2015 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2015). IEEE.

[13] Jiang, S., Chen, Y.: Hand gesture recognition by using 3dcnn and lstm with adam optimizer. In: Pacific Rim Conference on Multimedia, pp. 743– 753 (2017). Springer.

[14] Liu, T., Zhou, W., Li, H.: Sign language recognition with long short-term memory. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 2871–2875 (2016). IEEE.

[15] Liu, Z., Chai, X., Liu, Z., Chen, X.: Continuous gesture recognition with hand-oriented spatiotemporal feature. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 3056–3064 (2017).

[16] Hu, X., Tan, L., Zhou, J., Ali, S., Yong, Z., Liao, J., Liu, L.: Recognizing chinese sign language based on deep neural network. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4127–4133 (2020). IEEE.

[17] Liang, Z.-j., Liao, S.-b., Hu, B.-z.: 3d convolutional neural networks for dynamic sign language recognition. The Computer Journal 61(11), 1724– 1736 (2018).

[18] Barve, P., Mutha, N., Kulkarni, A., Nigudkar, Y., Robert, Y.: Application of deep learning techniques on sign language recognition – A survey. Data Management, Analytics and Innovation, 211–227 (2021).

[19] Robert, Y., Nigudkar, Y., Kulkarni, A., Mutha, N., Barve, P.: Literature survey: Application of machine learning techniques on static sign language recognition. In: International Conference on Innovations in Bio-Inspired Computing and Applications, pp. 179–186 (2020). Springer.

[20] Patil, A., Kulkarni, A., Yesane, H., Sadani, M., Satav, P.: Literature survey: Sign language recognition using gesture recognition and natural language processing. Data Management, Analytics and Innovation,

197–210 (2021).

[21] Tyagi, A., Bansal, S.: Feature extraction technique for vision-based indian sign language recognition system: A review. Computational Methods and Data Engineering, 39–53 (2021).

[22] Varsha, M., Nair, C.S.: Indian sign language gesture recognition using deep convolutional neural network. In: 2021 8th International Conference on Smart Computing and Communications (ICSCC), pp. 193–197 (2021). IEEE.

[23] Gupta, R., Golaya, S., Srinivasan, R.: Transfer-learning based userpersonalization of indian sign language recognition system. In: 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 615–620 (2022). IEEE.

[24] Sharma, S., Singh, S.: Recognition of indian sign language (isl) using deep learning model. Wireless Personal Communications 123(1), 671–692 (2022).

[25] Bahia, N.K., Rani, R.: Multi-level taxonomy review for sign language recognition: Emphasis on indian sign language. Transactions on Asian and Low-Resource Language Information Processing (2022).

[26] Pereira-Montiel, E., P´erez-Giraldo, E., Mazo, J., Orrego-Metaute, D., Delgado-Trejos, E., Cuesta-Frau, D., Murillo-Escobar, J.: Automatic sign language recognition based on accelerometry and surface electromyography signals: A study for colombian sign language. Biomedical Signal Processing and Control 71, 103201 (2022).

[27] Li, R., Meng, L.: Multi-view spatial-temporal network for continuous sign language recognition. arXiv preprint arXiv:2204.08747 (2022).

[28] Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-based sign language recognition without temporal segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018).

[29] Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding 141, 108–125 (2015).

[30] Tyagi, A., Bansal, S.: Hybrid fist cnn approach for feature extraction for vision-based indian sign language recognition. Int. Arab J. Inf. Technol. 19(3), 403–411 (2022).

[31] Mavi, A.: A new dataset and proposed convolutional neural network architecture for classification of american sign language digits. arXiv preprint arXiv:2011.08927 (2020).

[32] Triesch, J., Von Der Malsburg, C.: Robust classification of hand postures against complex backgrounds. In: Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, pp. 170–175 (1996). IEEE.

[33] Pisharady, P.K., Vadakkepat, P., Loh, A.P.: Attention based detection and recognition of hand postures against complex backgrounds. International Journal of Computer Vision 101(3), 403–419 (2013).

[34] Berg, A., Deng, J., Fei-Fei, L.: Large scale visual recognition challenge 2010 (2010).

[35] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). IEEE.

[36] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[37] Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research 12(7) (2011).

[38] Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012).

[39] Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. In: Neural Networks: Tricks of the Trade, pp. 437–478 (2012). Springer.