International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I6P204
Volume 70 Issue 6, 32-41, June 2022

Original Article

A Deep Learning Framework for Real-Time Sign Language Recognition Based on Transfer Learning

Vijeeta Patil[1], Sujatha C[2], Shridhar Allagi[3], Balachandra Chikkoppa[4]

[1,3,4]*Department of Computer Science and Engineering, KLE Institute of Technology, Hubballi.*
[2] *School of Computer Science and Engineering, KLE Technological University, Hubballi.*

vijeetapatil@gmail.com

Received: 16 March 2022 Revised: 05 May 2022 Accepted: 22 May 2022 Published: 27 June 2022

Abstract - *Hearing Impairment is common among all the groups irrespective of gender, location, and genes. The cause of it may vary according to the geographical and biological aspects. However, for the betterment of humankind, the solution to this is obvious, either through medical or with the help of technology. Sign Language recognition is a worldwide concern across the globe. The use of technology has a scope in aiding the necessary help in recognition of sign language. The major challenge lies in detecting and understanding signs, as the language differs across the various geographical regions, and there are no specific rules for understanding them. Hence, this research article uses a transfer learning algorithm with TensorFlow object detection to recognize the sign language. The proposed model has achieved an accuracy of around 97.87% for different types of sentences used in the experimentation. The main advantage of the proposed model is that it is feasible to use different sign languages, such as American sign language and any regional sign language. The system is helpful to the deaf and dumb community's society and encourages such people's upliftment.*

Keywords - *Machine Learning, Transfer Learning, Sign Language, Object Recognition, and American Sign Language.*

1. Introduction

The Language of Sign is a form of interaction among differently-abled people with speaking and listening/hearing impairments. These people use sign language to communicate and express themselves to others. Other people who can listen and speak find it challenging to understand the gestures of the sign language. This language uses fingers, hand signals, gestures, and expressions of the face. Body language movements are added to express the information to others. Sign language serves as a means of communication for the differently-abled community and those suffering from apraxia of speech, Down syndrome, and cerebral palsy. There are no specific rules for sign language because, in different parts of the world, a group of people found a unique way to communicate with deaf people. There are nearly 130 to 300 types of sign language. Each country has its own, such as ISL-Indian Sign Language, ASL-American Sign Language, Auslan-Australian Sign Language, and BSL-British Sign Language. Hence, sign languages are not universal. This problem can be solved by HCI (Human-Computer Interface); sign language recognition is a chunk among HCI applications. To bridge the gap between the differently-abled community and the hearing world, it is necessary to convert sign language into text or speech, making communication easy for both worlds. This conversion can be done with the help of deep learning techniques and machine learning algorithms. There are two types of sign language: fingerspelling and alphabet representation in sign language. In the latter method, a combination of alphabets denotes words in the respective language. In this work, we process the sign language captured in real-time to recognize the sign language words.

According to a survey in the year 2021 conducted through the Census India web portal [15], the total percentage of the differently-abled people in the population is 2.1% of the total population, as depicted in Table 1. Many different machine learning and deep learning methods, such as decision trees, random forests, support vector machines, and artificial neural networks, can detect a sign language—the semantic segmentation methods, such as U-Net with ResNet 101.

Table 1. Population of hearing and speaking impairments for the year 2021

Population in India	Count
Total population	1.39 billion
The population of hearing impairments	63 million
The population with speaking impairments	21 million

The proposed work detects the sign language of a differently-abled person and processes them into sentences to make people's interactions easy for both parties. We suggest using the transfer learning algorithm with TensorFlow object detection to recognize the sign language. The videos are captured, processed, and recognized by the system for real-time analysis as well as static images can also be detected, including 5 to 6-word long sentences. The system does the processing for alphabets and numbers of pictures. The proposed model has achieved an accuracy of around 97.87% for different types of sentences described in other sections. Another advantage of the system is that it is feasible to use different sign languages, such as American sign language and any regional sign language. The system is helpful to the deaf and dumb community's society and encourages such people's upliftment. The research article is organized into six sections. The first section gave a comprehensive need for sign language detection. Section two gave the state-of-the-art review using variously supervised and unsupervised models. Sections three and four gave the comprehensive methodology work and resulted in discussions. The last section briefed the future scope.

2. Literature survey

Mohammed et al. (2022). The Microsoft Kinect sensor was used to find and separate the hand area in a depth image. When the colors of the skin and hands match up with the face, this method is excellent. In this case, convolutional neural networks (CNN) are used to generate ISL features automatically. The model resulted in 99.3% accuracy, and the method can read ISL alphabets correctly in real-time.

Joudaki & Rehman (2022) proposed a geometric neural network model to recognize the sign language alphabet. The sign language alphabet is used for communication. These cameras capture hand movements. So, its deep skills are used. GSLR stands for Geometric Sign Recognition System. Simplistic design has various applications. This method exploits qualities that are consistent regardless of hand movement. Both features improve accuracy. Precise hand movements can improve a neural network's recognition accuracy. It may also recognize the baby's sign language.

Sharma et al. (2021) Static recognition of Indian sign languages was investigated using deep learning models, gradient-based optimizers, and optimization hyperparameters. The best number and alphabet recognition accuracy were 99.0 percent and 97.6 percent, respectively, using a publicly accessible ISL dataset and a custom three-layered CNN model. The ISL dataset correctly detected integers 96.2 percent of the time and alphabets 90.8 percent, outperforming other pre-trained models.

Sharma & Singh (2021). Based on movement, convolutional neural networks recognize sign language. This model outperforms past CNN approaches due to its fewer parameters and straightforward representation. Additionally, they used VGG-11 and 16 to evaluate the model's performance. Two datasets were used to assess performance. ISL motions were collected using an RGB camera, followed by an ASL dataset. The accuracy of ISL datasets is 99.96%, while ASL datasets are 100 percent correct. It is contrasted with contemporary methods. Numerous factors contributed to the resilience. The suggested technique outperforms existing methods in describing enormous volumes of motions. This data set is invariant to rotation and scale.

Wadhawan & Kumar (2020) The study uses deep learning convolutional neural networks (CNN) to represent robust static signs in sign language identification (SLI). This study collected 35,000 photos of 100 fixed characters from various users. The proposed strategy is evaluated using roughly 50 CNN models. Using more optimizations, the suggested method achieved 99.72% training accuracy on colored images and 99.90% on grayscale—precision, recall, and F-score measures the proposed system's performance.

Kolivand et al. (2020) The ASLNN framework uses a neural network that can read geometrical hand input to read the American sign language alphabet. The hand is segmented based on the depth analysis results—depth-based geometrical sign language recognition (DGSLR). The DGSLR used a less complex hand segmentation method. Using the discrete cosine transform and the moment invariant improves geometric feature extraction. The iteration found that adding derived characteristics enhanced accuracy rates. Then an ANN secures the required outcomes (ANN). According to another magazine's research, the ASLNN can identify your hands 96.78%.

V., A., & R., R. (2020) The proposed model uses deep convolutional neural networks (CNNs) to recognize hand motions, which are an essential part of the sign language lexicon (SLL). Researchers discovered that the strategy outperformed previous strategies on two publicly available datasets (National University of Singapore hand posture and American fingerspelling A). Hand-crafted feature descriptors obtained from motion images are used in current vision-based recognizers. In chaotic and complex conditions, the application of such systems is limited.

Abdulhussein & Raheem (2020) The application of deep learning to recognize static American Sign Language motions is discussed. It is first scaled using static ASL binary pictures from Bicubic. Effective border hand recognition is also feasible using the Robert edge detection approach. The

ASL's 24 alphabetic static letters are classified using convolutional neural networks and deep learning. The classification accuracy is 99.3 percent, and the loss function error is 0. 0002. In 36 minutes, it took 15 seconds and 100 iterations. This method is fast and effective compared to CNN, SVM, and ANN-based training methods.

Al-Hammadi et al. (2020) proposed a dynamic hand gesture recognition system incorporating deep learning for hand segmentation, local and global feature representations, globalization, and sequence recognition. The recommended method was evaluated using 40 dynamic hand gestures performed by 40 people in an uncontrolled environment. The results show that the recommended method outperforms existing procedures, demonstrating effectiveness.

Gangrade & Bharti (2020) proposed a technique that works best when the background is a different color than the skin, and the hand is close to the face. Convolutional neural networks (CNNs) automatically extract features from Indian sign language (ISL) data. Rotation and size have minimal effects on these features. The proposed approach accurately recognizes gesticulations 99.3% of the time.

Ahuja et al. (2019) The authors collected real-time images of 500 unique hand movements using a camera. After photographing them, they were able to get an accuracy rate of 87.2% using their method. Convolutional Neural Networks (CNNs) with the twenty-four hand signs of American Sign Language (ASL) promote intercultural communication (ASL). Other tasks, like image processing, necessitated the usage of OpenCV. CNN claims that when it makes predictions using the kaggle.com database, the network's forecasts are 99.7% correct.

Ahuja et al. (2019) The authors created skeleton sequences using four subnetworks and three spatiotemporal feature classifications. These skills were used to acquire and classify skeleton sequences. It is called "late feature fusion" because it combines features from many networks into one fusion classifier. That is, each subnetwork only recognizes joint motions. Optimized weighted ensemble (OWE) combines them all. Hand gestures may now be accurately recognized with a new deep-learning network design.

Darwish et al. (2016) The model employed Type-2 Fuzzy HMMs to determine if the hand continued to move (T2FHMM). Singular Value Decomposition is used to visualize the geometric structure, such as a grid (SVD). It is a fantastic method of reinforcing and rewarding positive behavior. SVD enhances the appearance of matrices. The SVD method has difficulty with processing data with many hand motion attributes. It is called the Eigen decomposition. Standard HMM arithmetic operators can be used in place of Type-2 fuzzy operators. T2FHMMs are capable of accounting for random and vague uncertainty in serial data. T2FHMMs excel at various tasks, including noise reduction in hand gestures and languages. The suggested technique correctly detected 100% homogeneous hand shots, and 95% blocked hand shots.

Kharate and Ghotkar’s (2015) model used Regulated time dilation that aids in deciphering Indian sign language. In comparison to rule-based word recognition, Dynamic Time Warping is more flexible. Inverted indexing complicates the process of identifying continuous standard sentences. In ISL sentence synthesis, inverted indexing ignores grammatical symbols and sentence structure. ISL translated a limited number of words. A sentence or keyword list may assist in translating an English term or phrase. Regardless of the language used, the concepts were comparable.

According to a survey in the year 2021 conducted through the Census India web portal [15], the total percentage of the differently-abled people in the population is 2.1% of the total population, as depicted in Table 1. Many different machine learning and deep learning methods, such as decision trees, random forests, support vector machines, and artificial neural networks, can detect a sign language—the semantic segmentation methods, such as U-Net with ResNet 101.

3. Dataset Description

A newly created dataset of invited 50 people, including deaf and ordinary people, is collected. Each person performs actions for different sentences, nearly 15 in number, described in Table 2. The videos are recorded using an OpenCV webcam on a laptop or desktop with moderate configurations. The videos are shooted from different angles, with fast and slow momentum of the hands of the person. The distance between the user and the laptop, correctly positioned, is about 30 cm. This distance can be exceeded by 32 to 35 cm. Sample data sets are shown in figure 1 from the videos extracted via webcam from a laptop. Figure 2 consists of static images of sign language alphabets and numbers. A single alphabet and a number are captured with 1199 different snaps—total numbers covering 1199 snaps of digits between 0 to 9 and 26 alphabets with 1199 snapshots.

Actions	Sample 1	Sample 2	Sample 3	Sample 4
Hello				
I love you				
Yes				
No				
Thank you				

Fig. 1 Sample Datasets extracted from videos representing a few signs and language words

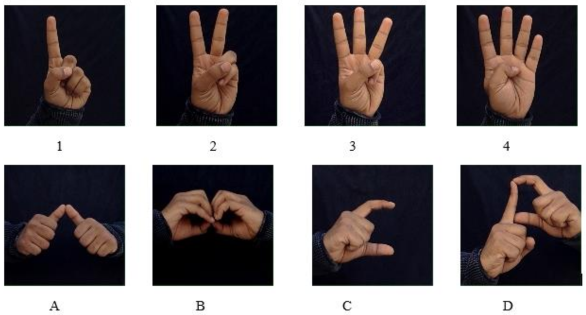

Fig. 2 Sample datasets for numbers and alphabets of static images

Table 2. Count of Training and Testing Data set

Total count of Persons	50	Actions performed are 10 in number
Training set	37	37x10 =370
Testing set	13	13x10=130

Figure 2 gives the snapshots of static actions performed for numbers and alphabets. These datasets serve as the input for training our model in recognizing the digits and alphabets. Table 2 gives the details of the number of training instances details.

4. Methodology

The sign language detection method is built by collecting images using OpenCV with Python by importing the OpenCV package in the library section, which automatically enables the camera of a desktop or laptop to start capturing the videos of the subject performing the sign language sentences using one hand or both hands. The system works in dual mode, i.e., double-handed gesture recognition. Figure 3 gives the details of the proposed model. Once the videos are collected from the subject, only images with the correct sign words are captured, and later the labeling of images is done for object detection.

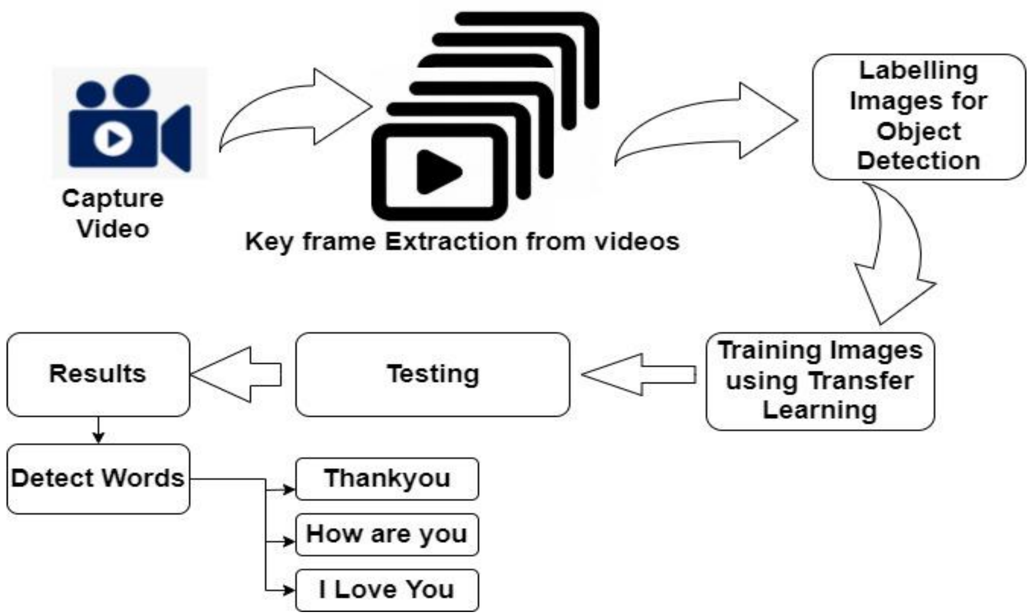

Fig. 3 Proposed Framework for Sign Language Detection system for real-time images

The following steps are incorporated in the initial stage of processing.

1. Capture images in real-time using a webcam with the help of OpenCV-python. The images extracted from videos are 480px with a 4:3 aspect ratio, 30fps, and around 50Hz flicker reduction.
2. Label the images using labeling, the annotation tool for graphics images, programmed in Python using Qt for the graphical interface. The photos are labeled with the words “hello,” “Hi,” “I love you, “thank you,” “Yes,” "No,” and “I don’t know,” “How are you, “I don’t,” “I am good.” About ten sentences, including 3 or 4 words in a sentence, are labeled. When we have a lot of different signs, we make detection boxes and label them with specific phrases
3. Once the labeling of images is completed, Update the label map. The annotations are saved in PASCAL VOC format and JPEG snapshots as XML files.
4. These images, collected now, can be divided into training and testing data sets. This dataset partition helps you overcome the fitting problem with the bounding boxes. The XML files of these images contain the height and depth of the image as mentioned in the bounding box.

5. The next step is to update the label map and generate a TensorFlow record. It is an object detection API and a graphical approach to data visualization. In our system, deep learning algorithms are compatible with the tensor flow as it helps to develop neural networks for detecting sign language.
6. Update the batch size of data based on the number of epochs by detecting the checkpoint and training the samples. Simultaneously, the training and testing TF records are created and check the updates are generated in the TF folder with the increased number of epochs and current time.
7. The Ssdmob net is used from a pre-trained model to match the existing configurations, increase the epochs from 5000 to 10000 and retrain the model again to find the exact accuracy value.
8. Now the model has been successfully trained, the loss metric for the first 100 steps is 0.752. The best results that we have acquired are around 0.18 to 0.15. Increase the count to 10,000, and the system loses about 0.99.
9. Hence, the results and discussions section describes real-time sign language detection with efficient accuracies for each sentence.

The architecture of the tensor flow recognition system is explained in Figure 4. It consists of three modules: data collection, tensor flow hub, and detection system to predict the results. In the first data collection module, videos are collected, and labeled images are obtained. In the second module, the tensor flow hub consists of Tf. Keras is an optimizer, image classification model Image net, and transfer learning are carried out with the pretrained SSD mobnet model, and the results are predicted as naming conventions for sentences such as “thank you,” “yes," "no,” etc.

4.1. A Deep Learning Framework: Transfer learning with the TensorFlow algorithm

1. The model tf. Keras is used by the tensor flow hub and depends on selecting the number of layers used in the detection system.
2. The next step is to utilize the image classification model named the ImageNet classifier, which classifies the images based on the hyperparameters of TensorFlow.
3. Perform simple transfer learning on the training and testing datasets separated in a 70:30 sample ratio.
4. Pre-train the samples using SSD mobnet, available in the tensor flow Hub as illustrated in figure 2.
5. The last step is to feed the system with testing samples that recognize the sentences trained in steps 3 and 4.
6. The evaluation metrics obtained, such as training loss, testing loss, epochs count, and batch size, must be substituted in equation 1 to get the optimized parameter for accurate prediction.

$$Q=inv(A\_T)*A\_T*P \qquad \ldots(1)$$

Where,
A is the matrix of our training data
P is the output value
A_T represents the transpose of the matrix A

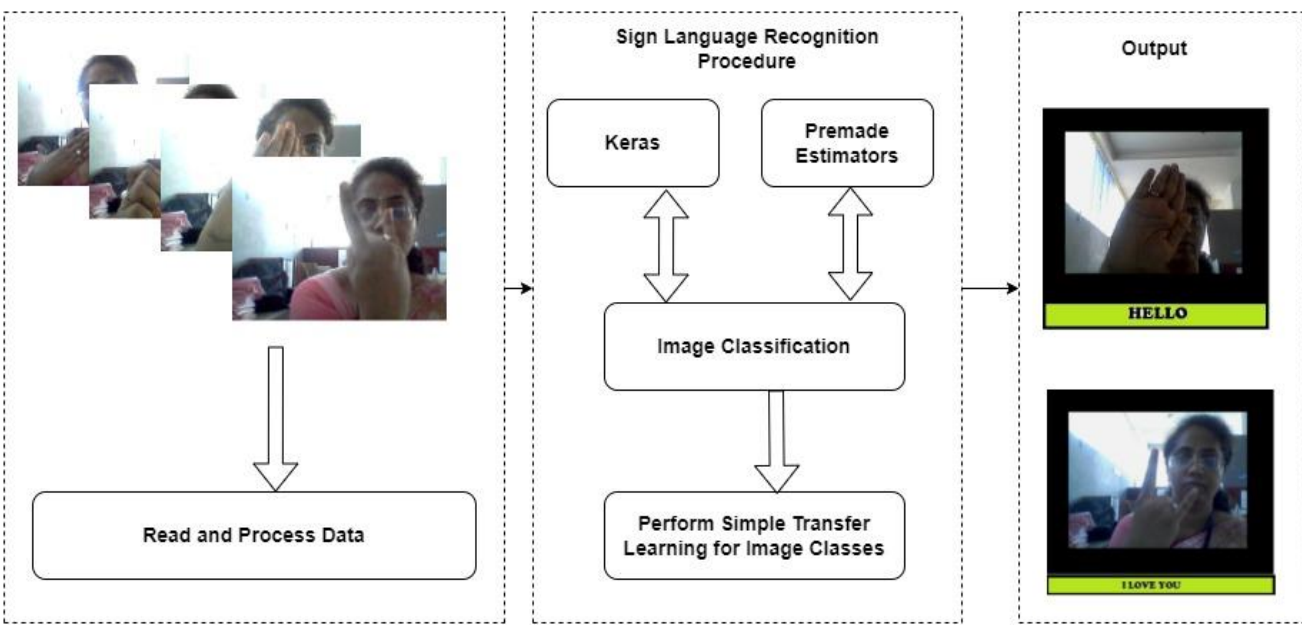

Fig. 4 Architecture for TensorFlow for recognition of sign language

4.2. Extraction of keyframes from videos

A video consists of continuous processing data where the extracted frame from this video can be any gestured frame or non-gestured frame. The edges are segmented with the gradient method, which extracts several keyframes from a video sequence. Consider a sentence such as “how was your day?”. The gesture recognition starts from the 30th numbered frame and ends between the 150th and 180th frames, as illustrated in figure 5.

Fig. 5 Extraction of keyframe of the sentence “how was your day?”

4.3. Feature Extraction in recognition of sign language architecture

The principal component analysis and orientation histogram are applied to the frames extracted from the previous step. The following procedure is carried out:

1. Consider the subsample frame or the image of a 640x480 image being converted into a 60x40 size image, thereby reducing the storage complexity and boosting the processing time.

2. The value of the gradient is calculated direction-wise, such as
   $Dir(a)=(q(s+1,x)-q(s-1),y))/2$ …(2)
   Where $q(x,y)$ denotes an Intensity function at the positions x and y.

3. Find the magnitude $m(x,y)=\sqrt{(x^2+y^2)}$

This way, we will calculate each frame's magnitude and direction and use it in the next stage.

5. Results and Discussions

Training the TensorFlow models was segmented into 75:25 samples of training data samples and a testing dataset. The TensorFlow model is trained with the following hyperparameters, such as

- A count of 5000 for each sentence and an incremented count of 10,000 for the next sentence
- Batch Size:40
- Number of layers=3,5
- learning rate=0.keyframe1
- Optimizers=tf.Keras
- Initializers =tf. Keras. Initializers.

The experiments were conducted on nearly 15 sentences, each consisting of 4–5 words. It depends on the punishment for having 5 to 6 gestures. Each action of the deaf or hard of hearing person is recorded 25 times by changing the epoch count. This count of 25 is split into 17 times for training and the remaining eight times for testing. The real-time testing was conducted on nearly 50 people within 15 to 50 years.

The ImageNet classifier has the following results compared with other distance-based classifier algorithms: The ImageNet classifiers of tensor flow hold the possible results better than different classifiers. Table 3 depicts the results.

Table 3. Compares ImageNet classification results to Euclidean distance and chessboard distance classifiers

The phrases formed by Deaf and Dumb Person	ImageNet Classification of TensorFlow	Euclidean Distance Classifier	Chessboard Distance Classifier
“Thank You”	97.02%	95.6%	90.56%
“How are You”	96.98%	96.5%	90.78%
“How was your Day”	98.56%	94.8%	91.9%
“I love You.”	97.8%	95.8%	91.87%
“Yes”	96.89%	94.4%	90.67%
“No”	98.9%	93.6%	93.0%
“Are you Listening”	99.7%	92.89%	92.89%
“Come and meet me”	98.54%	95.89%	93.09%
“I don’t”	97.43%	97.9%	93.67%
“I am Good”	96.78%	93.8%	93.78%
“hi”	98.87%	94.8%	93.67%
“Hello”	99.09%	93.3%	92.19%
“I am reading”	98.90%	93.6%	91.89%
“Can I talk to you”	97.89%	95.8%	92.65%
“Welcome”	99.09%	92.7%	92.67%

The proposed sign language detection system is compared with the existing system based on the techniques and the number of sentences and alphabets detected with accuracy rates. The models can also be used in different sign languages, but the proposed system can recognize other sign languages, as in Table 4.

Table 4. Comparison of the proposed method with the state-of-art methods on the ISL dataset.

Ref	Techniques used	Gestures recognized	Accuracy	Compatibility with other languages	Static/Dynamic Videos /Images
[8]	Extraction of critical points, Scale Invariant feature transform	All 26 English alphabet	91.11%	No	Only static images
[9]	Adaptive Matching	The number of persons is ten, and sentences are 20	96%	No	Both static and dynamic images but no videos
[10]	Hidden Markov model and keyframe Extraction	The number of persons is 5, and the sentences are 10	95%	No	Static/Dynamic images and videos
Proposed work	Tensor flow Detection API	The number of persons is 50, and the sentences are 15	97%	Yes	Static and dynamic images and videos

The number of epochs counted being trained for the first 5000 is executed and noted. The execution time and test loss are calculated by adding more epochs presented in figure 6. Then calculate the test loss shown in figure 6 with TensorFlow Keras optimizers, and the learning rate of the fine-tuning Keras optimizers is noted.

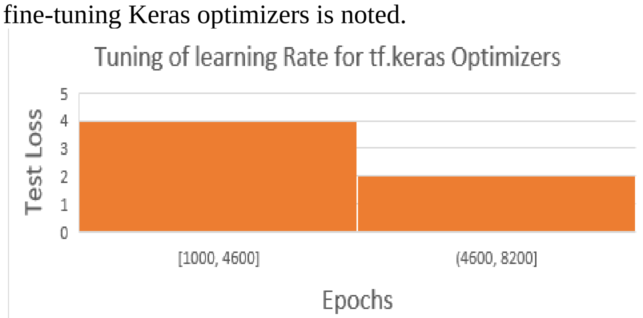

Fig. 6 Tuning of learning Rate of Keras optimizers

The results obtained are shown in Figure 7 with sentences formed by people. Maximum accuracy of 98.9% is achieved depending on the repeated number of actions performed by the person 10 to 15 times, and testing is performed.

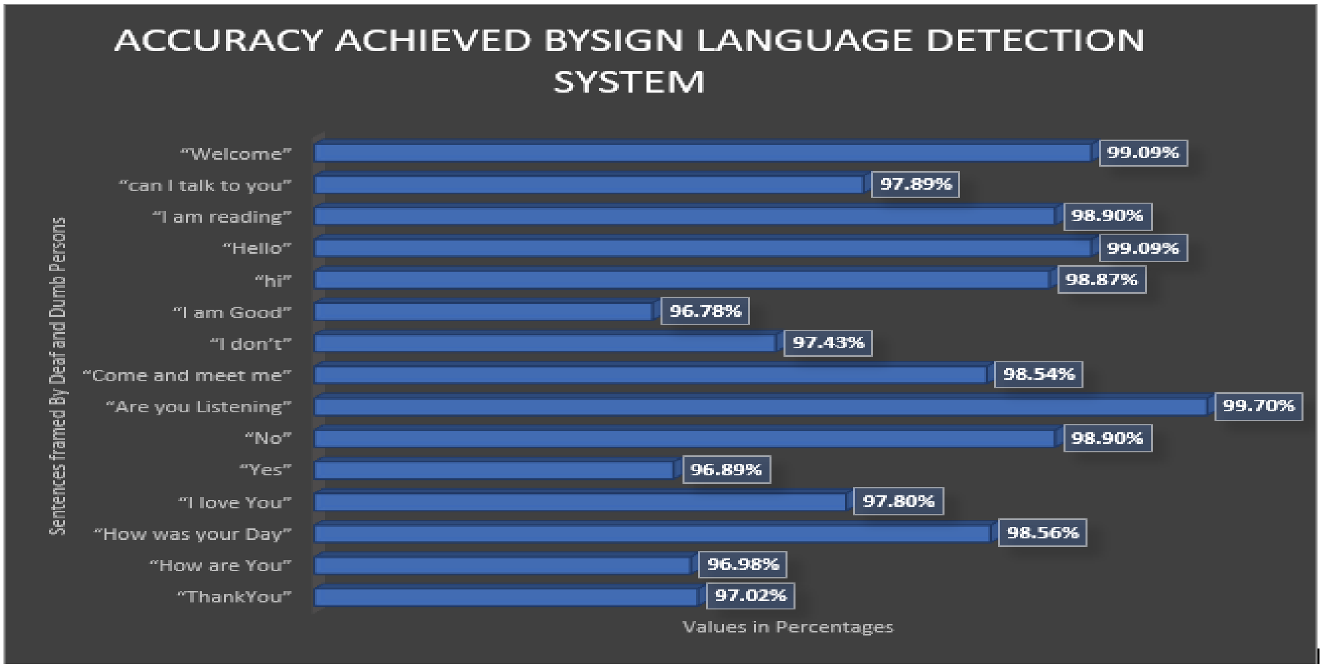

Fig. 7 Accuracy of the Proposed model

The values of the hyperparameters are obtained in Table 5 of tensor flow, such as train loss, test loss, training accuracy, validation accuracy, and batch size.

Table 5. Evaluation Metrics of Hyperparameters used in TensorFlow

Learning Rate	Train Loss	Test Loss	Training Accuracy	Validation Accuracy	Batch Size
0.1	0.12	0.99	96.99	89.90	30
0.2	0.05	0.98	97.02	92.89	35
0.5	0.06	0.97	97.89	94.87	40

The F1 score was calculated for the real-time performance of the system on training and testing datasets in the ratio of 70:30. A total of 50-person datasets are created on average. The f1 score of 0.9655 is obtained and illustrated in Figure 8.

The calculations are as below:

True Positive Rate: TPR = TP/((TP + FN)) …(3)

Specificity: SPC = TN/((FP + TN)) …(4)

Predicted Positive value PPV =TP/((TP + FP)) …(5)

Discovery rate for false: FPR = FP/((FP + TN))…(6)

False Negative Rate: FNR = FN/((FN + TP))…(7)

Accuracy: ACC = ((TP + TN))/((P + N)) …(8)

F1 Score = 2TP/((2TP + FP + FN)) …(9)

Where

TP=True Positive value,

FN= False Negative Value,

FP=False Positive value,

TN=True Negative Value

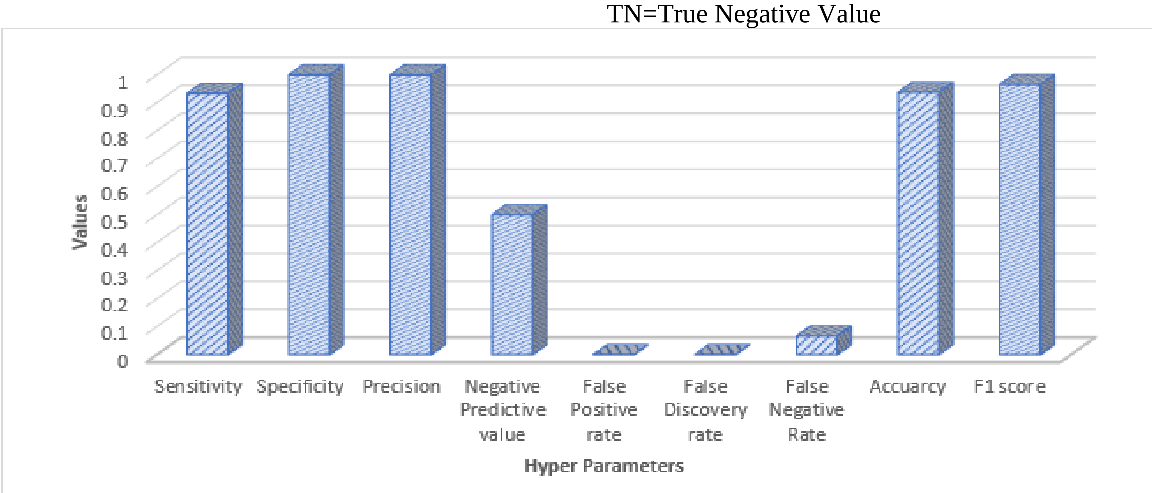

Fig. 8 Hyperparameter values computed for the proposed model

7. Conclusion

The various advancements in machine learning and computational methods are diverting the focus of several researchers associated with medical innovations in aiding the help to needy. The proposed model using transfer learning has given the optimum results in recognition of sign language in real-time and achieved an accuracy of up to 98%. The proposed model used the samples collected from 50 persons in static and dynamic modes using the frames of video segments. The model can perform more optimum with the usage of hybrid models. In the future, the model can be iterated with more training data, and also, a subset of signs can be increased, leading to a self-help device for blind, impaired persons.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] V. A. & R. R, A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition, Procedia Computer Science. 171 (2020) 2353–2361. https://doi.org/10.1016/j.procs.2020.04.255

[2] Sharma P & Anand R. S, A Comprehensive Evaluation of Deep Models and Optimizers for Indian Sign Language Recognition, Graphics and Visual Computing. 5 (2021) 200032. https://doi.org/10.1016/j.gvc.2021.200032.

[3] Wadhawan A & Kumar P, Deep Learning-Based Sign Language Recognition System for Static Signs, Neural Computing and Applications. 32(12) (2020) 7957–7968. https://doi.org/10.1007/s00521-019-04691-y.

[4] Kolivand H, Joudaki S, Sunar M. S & Tully D, A New Framework for Sign Language Alphabet Hand Posture Recognition Using Geometrical Features through Artificial Neural Network Part 1, Neural Computing and Applications. 33(10) (2020) 4945–4963. https://doi.org/10.1007/s00521-020-05279-7.

[5] Ahuja R, Jain D, Sachdeva D, Garg A & Rajput C, Convolutional Neural Network Based American Sign Language Static Hand Gesture Recognition, International Journal of Ambient Computing and Intelligence. 10(3) (2019) 60–73. https://doi.org/10.4018/ijaci.2019070104

[6] Mohammed A. A. Q, Lv J, Islam M. S & Sang Y, Multi-Model Ensemble Gesture Recognition Network for High-Accuracy Dynamic Hand Gesture Recognition, Journal of Ambient Intelligence and Humanized Computing. (2022). https://doi.org/10.1007/s12652-021-03546-6

[7] Ghotkar A. S & K. Kharate D. G, Study of Vision-Based Hand Gesture Recognition Using Indian Sign Language, International Journal on Smart Sensing and Intelligent Systems. 7(1) (2014) 96–115. https://doi.org/10.21307/ijssis-2017-647.

[8] Abdulhussein A & Raheem F, Hand Gesture Recognition of Static Letters American Sign Language ASL Using Deep Learning, Engineering and Technology Journal. 38(6) (2020) 926–937. https://doi.org/10.30684/etj.v38i6a.533.

[9] Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Bencherif M. A, Alrayes T. S, Mathkour H & Mekhtiche M. A, Deep Learning-Based Approach for Sign Language Gesture Recognition with Efficient Hand Gesture Representation, IEEE Access. 8 (2020) 192527–192542. https://doi.org/10.1109/access.2020.3032140.

[10] Gangrade J & Bharti J, Vision-based Hand Gesture Recognition for Indian Sign Language Using Convolution Neural Network, IETE Journal of Research. (2020) 1–10. https://doi.org/10.1080/0338342772063.2020.18.

[11] Ghotkar A & Kharate G, Interpretation Algorithm for Indian Sign Language Using Microsoft Kinect Sensor, JouDynamic Hand Gesture Recognition for Sign Words and Novel Sentence International of Pattern Recognition Research. 10(1) (2015) 24–38. https://doi.org/10.13176/11.626.

[12] Joudaki S & Rehman A, Dynamic Hand Gesture Recognition of Sign Language Using Geometric Features Learning, International Journal of Computational Vision and Robotics. 12(1) (2022) 1. https://doi.org/10.1504/ijcvr.2022.119239

[13] M. Darwish S, M. Madbouly M & B. Khorsheed M, Hand Gesture Recognition for Sign Language: A New Higher Order Fuzzy HMM Approach Hand Gesture Recognition for Sign Language: A New Higher Order Fuzzy HMM Approach, International Journal of Engineering and Technology. 8(3) (2016) 157–164. https://doi.org/10.7763/ijet.2016.v6.877.

[14] Sharma S & Singh S, Vision-Based Hand Gesture Recognition Using Deep Learning for the Interpretation of Sign Language, Expert Systems with Applications. 182 (2021) 115657. https://doi.org/10.1016/j.eswa.2021.115657.

[15] [Online]. Available: https://www.populationu.com/india-population.