

# Transforming Sign Language into Text using Machine Language

Surekha M<sup>1</sup>, Dr.Vineeta Chemchandani<sup>2</sup>, Mahesh G<sup>3</sup>, Samyukta Varshney<sup>4</sup> and Ishu Singh<sup>5</sup>

<sup>1-5</sup>JSS Academy of Technical Education, Noida. UP

Email: surekha@jssaten.ac.in, vkhemchandani@jssaten.ac.in, mahesh@jssaten.ac.in, samyuktavarshney@gmail.com, ishusingh5206@gmail.com

**Abstract**—The project "Transforming Sign Language to Text using Machine Learning" provides a technological communication bridge for those with hearing and speech impairments. There aren't many ways for people with speech and hearing impairments to communicate with others. Using sign language as a means of communication is one of these choices. Numerous technological advancements and much study have been made to help the blind and deaf. In this sign language identification paper, we have created a sign detector that can easily distinguish between the letters of the alphabet, numerals from 1 to 10, and a wide range of other signals and hand motions. The purpose of this research is to enable individuals to detect hand signals and gestures. To train a model for sign language translation, we will use a gesture recognition model. We have picked CNN as our learning model since it provides better accuracy than the other approaches, which will help people converse with others who are naturally hard of hearing and speech. If the epochs are taken between 1 and 10, we will have the optimal curve; otherwise, it may result in underfitting or overfitting. As the number of epochs grows, the number of times the weights in the neural network are modified. Optimal Curve and 100% accuracy are provided for 1 to 10 epochs.

**Index Terms**— Sign language recognition, Convolution Neural Network, Hand Gesture recognition, Machine Learning.

## I. INTRODUCTION

For those with hearing and speaking impairments, conversation is difficult. Those who communicate via sign language, which is difficult for most people to understand. For deaf, dumb persons, this causes problems or makes communication more difficult. Giving people the ability to detect hand gestures and signs is the aim of this initiative. We'll train a model for sign language conversion using a gesture recognition model. People will find it simpler to converse with those who are naturally hard of hearing and speaking as a result. In addition to CNN, a number of other techniques, such as KNN, LR, Naive Bayes Classification, and Support Vector Machine, can be used for this project. This endeavor may be seen as integration and bending to the expectations of the fortunate community because it will enable the disabled community to become even more integrated into the abled world. Hearing people's efforts to make accommodations for deaf persons may be diminished. The data gathering should be adaptable enough to include individuals with all skin tones and in all circumstances. A statistical bias could disadvantage deaf people from a specific cultural group. People with hearing loss, especially those in the hearing world, experience sadness and dejection more frequently. Communication

difficulties between the hearing and the deaf present great challenges and significantly lower quality of life. A machine-based, entirely sign language translator might be essential in the current scenario. This work shows how a variety of gesture-based communication signs may be correctly identified using convolution neural architectures. Even though this sector has made considerable progress, much work still needs to be done.

CNN was chosen because it offers a higher level of accuracy than the other methods. A computer programme that trains a model using the CNN method is made using the programming language Python. The system can recognize hand motions by comparing the input with an existing dataset built using American Sign Language. We will be able to translate Sign Language into text as an output so that consumers may recognize the signs used by sign language speakers. This idea is used in Jupiter Lab, a platform addition to the Anaconda documentation. The proposed system consists of four modules: hand segmentation, feature extraction, sign recognition, sign-to-text, pre-processing. With the use of image processing, segmentation is achievable. Among the features that are extracted are the Eigen values and Eigen vectors, which are used in recognition. Using the Linear Discriminate Analysis (LDA) method, the gesture was identified, and it was then converted into text and audio. The proposed system helps to reduce dimensions.

## II. RELATED WORK

Researchers have identified a variety of hand gestures using a variety of techniques that were used in a variety of fields. A variety of hand motions were recognized using methodologies that relied on vision, data gloves, soft computing techniques including artificial neural networks, fuzzy logic, genetic algorithms, and others like PCA and canonical analysis, among others.

TABLE I. LITERATURE SURVEY

| System                                                                                 | Feature Extraction                                                                                   | Methodology Used                                                               | Advantages                                                                                   | Disadvantages                                                                                                                    |
|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| Myasoedova<br>M.A Myasoedova<br>Z.P., Farkhadov<br>M.P. 2021 [1]                       | Sign Writing System.<br>Spatio-temporal form<br>Of sign.                                             | SW System<br>Distant Learning<br>Sign Writing.                                 | More accurate<br>recognition with the<br>use of Sign notation<br>and sign<br>writing         | Need to learn Signnotations for<br>new users.                                                                                    |
| Necati Cihan Camgoz,<br>Oscar Koller,<br>Simon Hadfield and<br>Richard Bowden<br>2020] | Sign Language<br>Translation CSLR.<br>Recognition<br>of Glosses.                                     | Connection<br>Temporal<br>Classification.                                      | Resolve problem on<br>doubling performance<br>of Previous system.                            | Extensive use of hardware's like<br>Transformers.                                                                                |
| Galib Ibne Haidar,<br>Hasin Ishraq<br>2020 [3]                                         | English Alphabet<br>Detection, Image<br>processing,<br>Data Glass.                                   | Convolution<br>Neural Networks.                                                | Meaning of different<br>hand gestures is<br>accurately<br>Interpreted. Accuracy<br>is 99.8%. | Just focused on identifying the<br>features of sign language. No<br>conversion of hand gestures into<br>Text or voice.           |
| Mariappan, H.M.,<br>et al<br>2019 [4]                                                  | Used hand gestures<br>and facial<br>expressions for the<br>identification of<br>Indian Sign Language | fuzzy c-means<br>clustering machine<br>learning technique.                     | Beneficial for those<br>who deaf or hard of<br>hearing.                                      | Only Identification ISL. Gap of<br>text Conversion.                                                                              |
| Yufei Yan, Zhijun Li,<br>Qunzhu Tao,<br>Chenyu Li<br>Rui Zhang 2019 [5]                | Gesture Recognition.                                                                                 | Kinect Sensor<br>DTW Algorithm.                                                | More than 90%<br>Recognition rate.<br>Improved recognition<br>Speed.                         | Complex Method.                                                                                                                  |
| Sandrine Tornay,<br>Marzieh<br>Razavi, Mathew<br>Magimai. Doss 2020<br>[7]             | Hand Shape Submit<br>Extraction Hand<br>Movement Detection.                                          | Kullback-Leibler<br>Divergence HMM                                             | Using Mobile<br>applications is easy<br>and feasible to use.                                 | The English language converted<br>into sign language. Only normal<br>people can understand the<br>language not the disabled ones |
| Kumar, A., et al 2016<br>[6]                                                           | Complicated single<br>hand movements.                                                                | HMM-based<br>approach for<br>recognizing a data<br>aligning technique<br>Used. | Stated approaches<br>are effective and<br>accurate.                                          | There is no accuracy showed.                                                                                                     |

The idea of a translation with skin tone to identify ASL using machine learning techniques has been put out by authors [7]. Authors have developed a system for segmenting skin tones that instantly shows the colour and associates a melody with it. They chose YCbCr because it is often used in video template code and creates useful colour tones

This paper [11] here, we're attempting to facilitate communication by converting the static and dynamic signs used in ISL into voice. The orientation of the hand and subsequent movements are designed with the aid of a glove-placed sensor and a flex sensor. Via wireless communication, which transforms it into additional speech output bits. In this experiment, they looked at LSTM networks that have built-up dependencies over time. As a result of this experiment, the 26 gestures could be recognized with a success rate of 98% accuracy.

### III. SYSTEM DESIGN

#### A. System Architecture

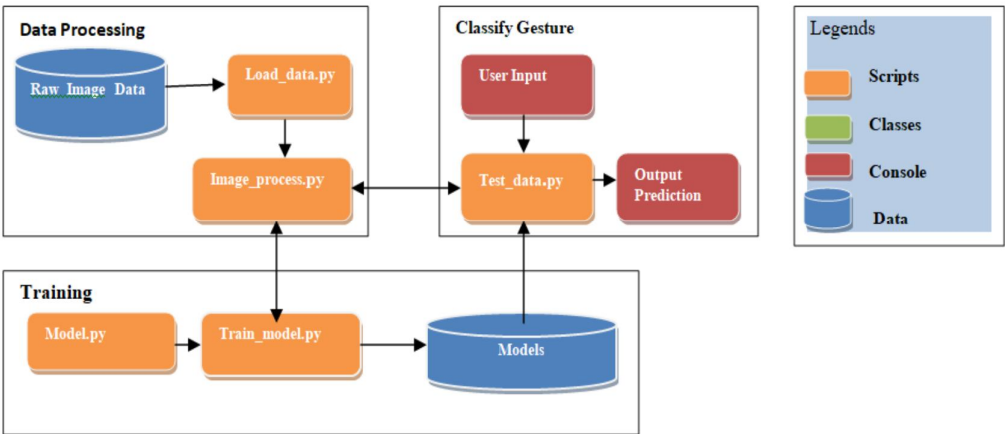


Fig 1. System Architecture

The Paper will be organized into three independent functional blocks, Data Processing, Training, and Classify Gesture, as indicated in Figure 2.

### IV. IMPLEMENTATION DETAILS

#### A. Creating the data set

For this project, we'll be creating our own data set. To get the live cam stream for the data set, we utilize OpenCV. Then, ROI will identify the hand gesture (Region of Interest). In order to achieve a high level of accuracy for sign recognition in a sign language recognition system, 10 photographs will be collected for every 26 signs. These images are part of the testing and training data. This is the initial phase of pre-processing. This article describes how a picture is sensed. Illumination is utilized to perceive a picture in order to acquire it. Scaling and other pre-processing techniques will also be applied. A database will be used to retrieve the image. When taking the picture. The signer makes adjustments to the far-off picture.

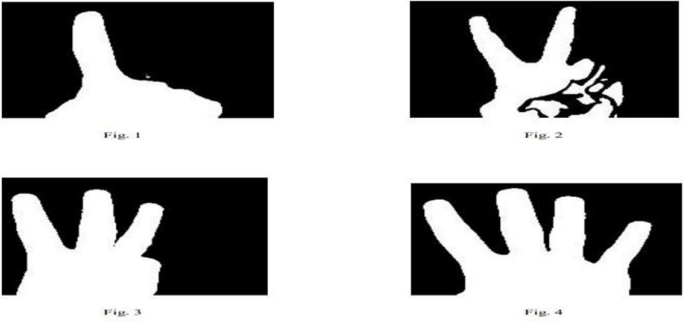


Fig.2

### B. Training the CNN

In order to import the train and test set data using the flow from directory function, we first load the data using Kera's Image Data Generator. The class names for the loaded images are taken from the names of the number folders. Create CNN immediately. We must now fit and save the model in order to use it from the previous module. Early halting callbacks and reduce LR on plateau are both used in training and both rely on the validation data set being lost.

For each unique period, the accuracy and loss are calculated using the validation data set. Reduce LR is used to lower the model's LR if the validation loss does not result in overshooting the loss minima.

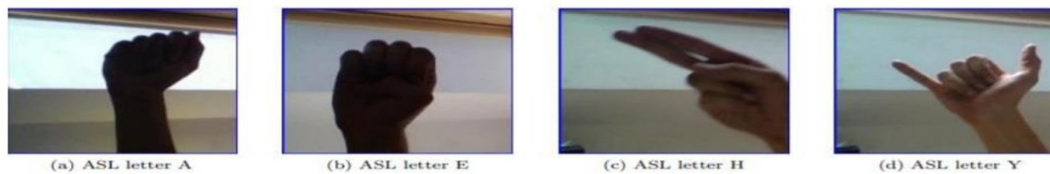


Fig.3



Fig.4

### C. Convolution Neural Network for Detection

A kind of neural networks known as CNNs is very effective at addressing computer vision issues. Beginning with lowlevel features, higher level features that are more sophisticated are gradually picked up by the starting layers.

#### D. The CNN Architecture functioning

The first convolutional layer is in charge of identifying low-level features like lines. Activation maps for each of the 16 2x2 filters in this layer have a size of 49x49. A rectifier linear unit (ReLU) layer is also added, which eliminates all negative values from the map and replaces them with a value of 0. By only taking into account maximum values in 2X2 portions of the map, applying a max-pooling layer decreases the activation to 25x25. Max-pooling layer, which searches for the greatest values in 3x3 areas of the map, further shrinks the activation map to 8x8x32. 64 filters of size 5x5 are used to decrease the input, producing an output of 4x4x64. A 64-element 1D array containing the compressed map is created. Using a dense layer, the map is enlarged to a 128-element array. A dropout layer eliminates random map elements to avoid overfitting the map. A dense layer reduces the map to an array of 44 elements, which stands for the number of classes.

#### E. LDA Approach

DK-dimensional matrix  $W$  is then used to map the samples into the new subspace of the greatest Eigen value matrix. (Where every column represents an eigenvector). In step two, create the scatter matrices (between-class and within-class scatter matrices).  $Y$  is an  $nd$ -dimensional matrix with an  $ith$  row representing the  $ith$  sample and  $X$  is a converted  $nk$ -dimensional. Matrix with the  $n$  samples projected onto the new subspace. These images are stored in a database, which can be created from scratch or from scratch.

#### F. Predict the gesture

We don't use the entire dataset all at once or one sample at a time. We employ a batch of training examples with a fixed quantity that is smaller than the actual dataset and refer to it as a mini-batch. By doing this, we are able to benefit from both of the previous alternatives' advantages. Consequently, we do the following actions in one epoch after producing the mini-batches of fixed size:

- Select a small quantity
- Provide it to the neural network
- Determine the mini-mean batch's gradient.
- Update the weights using the mean gradient we computed in step 3.

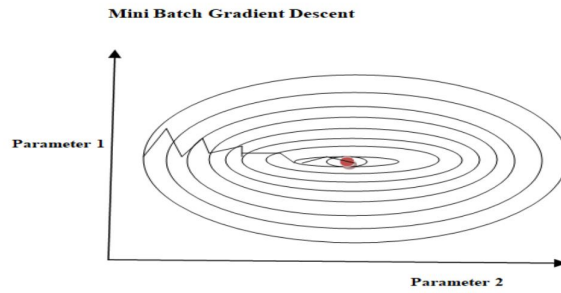


Fig.5

Repeating the first four stages for our mini-batches. The average cost of mini-batch gradient descent over the epochs fluctuates similarly to SGD because we are only averaging a small number of examples at a time.

## V. MODEL PERFORMANCE

Validation and Training Cross Entropy Loss and the Adam optimizer were used to train our models. So, we made the decision to concentrate on enhancing our tiny model. It had been trained originally to have a 40% validation accuracy. After converting the photographs into numpy arrays, we found that our laptops and desktop PCs could only manage 50-100 samples per class and 200 samples per class on our Google Cloud server, despite the fact that we had a very substantial dataset to deal with—3,000 examples for each of the 29 classes. We examined the effects of providing our models with extra data because we had to load minimal datasets. On our preliminary model, which used strides of 2 on both layers (instead of the strides of 1 and then 2, on our final model), we found the following link between samples/class and model accuracy:

TABLE II

| Layer | Samples/Class Average | Validation Accuracy |
|-------|-----------------------|---------------------|
| 1     | 25                    | 27.35%              |
|       | 50                    | 34.8%               |
| 2     | 100                   | 46.1%               |
|       | 200                   | 58.2%               |

This is probably because there aren't enough training samples, which results in poor generalization and sample space learning. Our model performed better when our dataset was expanded to 200 samples per class with a high validation accuracy of 60.3% in epoch.

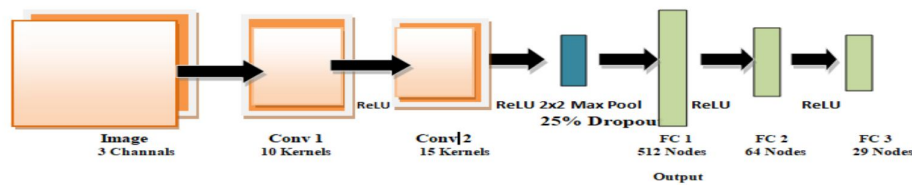


Fig.5 Machine learning Model

Through analysis of our losing function, the validation loss is increasing, which indicates that the model may have been overfit. When we applied ZCA whitening, augmentation, and filtering to our dataset, as shown in Figure 5, the model performance considerably increased. The maximum validation accuracy in epoch 24 was 77.25%. The plot of loss shows that the validation loss is still decreasing, but more slowly than the loss of training, demonstrating that the model is not overfitted excessively. This shows how using filters and ZCA whitening during image processing can improve the model's important characteristics.

## VI. TESTING

We tested our preprocessing on a test set made up of photos from the original dataset and our own image data collection to see if it really did produce a more robust model. Table 3 displays the models' performance on the

test set. We see that the model trained on preprocessed photos outperforms the model trained on the original images by a wide margin, most likely because the former exhibits less overfitting.

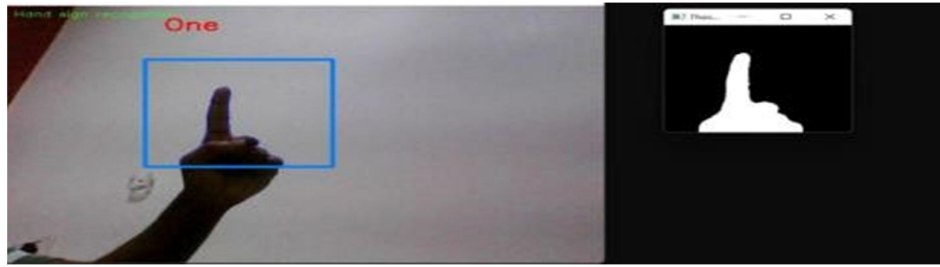


Fig.6

## VI. EXPERIMENT RESULT

One epoch is the number of times the neural network receives a whole dataset before and after it. We break an epoch into smaller batches because it would be too large to feed the computer all at once [10].

Table III

| No | Epoch | Iteration/Batch | Accuracy |
|----|-------|-----------------|----------|
| 1  | 1/10  | 151/151         | 96%      |
| 2  | 2/10  | 89/151          | 100%     |
| 3  | 3/10  | 91/151          | 100%     |
| 4  | 4/10  | 150/151         | 100%     |
| 5  | 5/10  | 150/151         | 100%     |
| 6  | 6/10  | 151/151         | 100%     |
| 7  | 7/10  | 77/151          | 100%     |
| 8  | 8/10  | 67/151          | 100%     |
| 9  | 9/10  | 153/151         | 100%     |
| 10 | 10/10 | 71/151          | 100%     |

We anticipate that the learning curve graphs will continue to improve until convergence. The model will overfit if we continue training it, and validation errors or loss start to rise [10]:

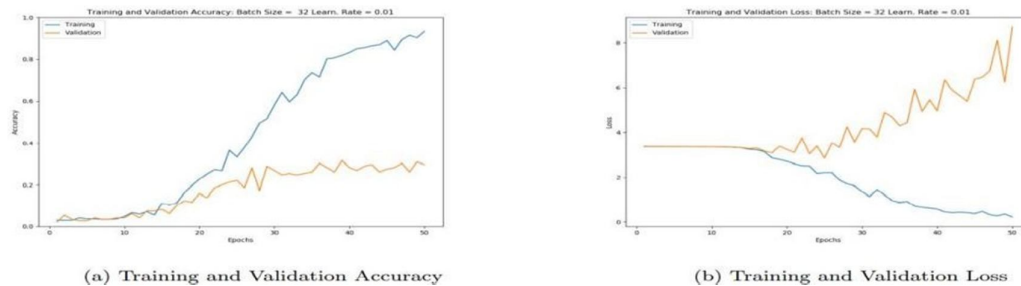


Fig.7

The weights in the neural network are adjusted more frequently as the number of epoch rises, and the curve changes from underfitting to optimal to overfitting. 100% accuracy and the Optimal Curve are given from 1 to 10.

## VII. CONCLUSION

In the current situation, the need for a machine-based completely sign language translator could be critical. This study demonstrates how convolution neural architectures can be used to precisely recognize numerous gesture-

based communication indicators. Even though there has been some progress in this area, there is still a lot of work to be done.

The most common places where this can be employed are public venues such as ticketing counters, hospitals, and schools. This can even be used to teach sign language to regular people via gestures and this can be used to recognize English alphabets. The machine's training set will be enlarged.

#### FUTURE WORK

The projected values from this model can be used in the Android application for the silent person to assist them communicate with others in the future. In addition to instantly converting American Sign Language into speech, this effort also entails anticipating the requirements of those who are deaf. As a result, a mute person might be more efficient and clear about their exact needs throughout the day. Future developments might make it possible for the deaf individual to obtain the forecasts on their phone and judge their accuracy.

#### REFERENCES

- [1] M. A. Myasoedova, Z. P. Myasoedova, and M. P. Farkhadov, "Multimedia technologies to teach Sign Language in a written form," 14th IEEE Int. Conf. Appl. Inf. Commun. Technol. AICT 2020 - Proc., 2020, doi: 10.1109/AICT50176.2020.9368720.
- [2] N. C. Camgöz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 10020–10030, 2020, doi: 10.1109/CVPR42600.2020.01004.
- [3] G. I. Haidar and H. I. Reefat, "Glove Based American Sign Language Interpretation Using Convolutional Neural Network and Data Glass," 2020 IEEE Reg. 10 Symp. TENSYP 2020, no. June, pp. 370–373, 2020, doi: 10.1109/TENSYP50017.2020.9230626.
- [4] H. Muthu Mariappan and V. Gomathi, "Real-time recognition of Indian sign language," ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc., pp. 1–6, 2019, doi: 10.1109/ICCIDS.2019.8862125.
- [5] Y. Yan, Z. Li, Q. Tao, C. Liu, and R. Zhang, "Research on dynamic sign language algorithm based on sign language trajectory and key frame extraction," 2019 2nd Int. Conf. Electron. Technol. ICET 2019, pp. 509–514, 2019, doi: 10.1109/ELTECH.2019.8839587.
- [6] A. Kumar, K. Thankachan, and M. M. Dominic, "Sign language recognition," 2016 3rd Int. Conf. Recent Adv. Inf. Technol. RAIT 2016, pp. 422–428, 2016, doi: 10.1109/RAIT.2016.7507939.
- [7] S. Shahriar et al., "Real-Time American Sign Language Recognition Using Skin Segmentation and Image Category Classification with Convolutional Neural Network and Deep Learning," IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2018-October, no. October, pp. 1168–1171, 2019, doi: 10.1109/TENCON.2018.8650524.
- [8] Y. Yan, Z. Li, Q. Tao, C. Liu and R. Zhang, "Research on Dynamic Sign Language Algorithm Based on Sign Language Trajectory and Key Frame Extraction," 2019 IEEE 2nd International Conference on Electronics Technology (ICET), 2019, pp. 509–514, doi: 10.1109/ELTECH.2019.8839587.
- [9] <https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.istockphoto.com%2Fillustrations%2Fsign-languagehands&psig=AOvVaw2vIP8ikxPATUCQ2etNAJOB&ust=1651730085513000&source=images-&cd=vfe&ved=0CAwQjRxqFwoTCKiPvZuUxf cCFQAAAAAdAAAAABAD>
- [10] E. Abraham, A. Nayak and A. Iqbal, "Real-Time Translation of Indian Sign Language using LSTM," 2019 Global Conference for Advancement in Technology (GCAT), 2019, pp. 1–5, doi: 10.1109/GCAT47503.2019.8978343.